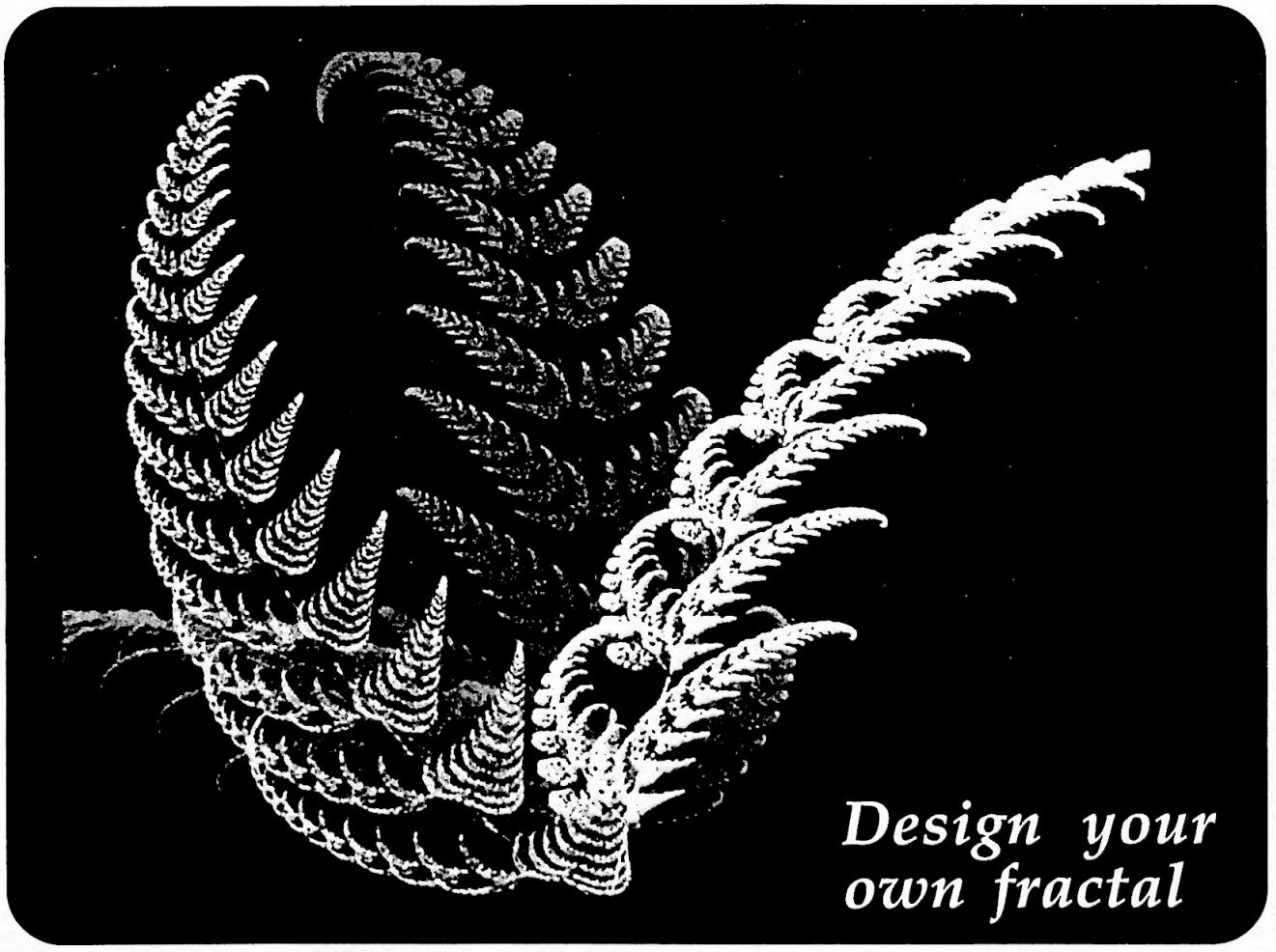
JANUARY / FEBRUARY 1990

Algorithm 1.2

THE PERSONAL PROGRAMMING NEWSLETTER



Computer Magic
Strange Attractors
The Homunculids

PROGRAMMING PLUS...

Introducing Software for Fractals!

THE DESKTOP FRACTAL DESIGN SYSTEM

Michael F. Barnsley

Designed to complement Michael Barnsley's best-selling text **Fractals Everywhere**, this software, which will run on IBM and compatible PCs with an EGA enhanced graphics board and 640K memory, will help to connect theoretical concepts with on-screen geometric modeling. Special features include interactive control of complex images, graphics generated by deterministic algorithms and by random iteration, a library of complex images, a library of JFS codes, and maps of parameter spaces and generalized Mandelbrot sets.

August 1989, $39.95/ISBN: 0-12-079063-7

Includes **The Desktop Fractal Design Handbook** (48-pages) and one floppy disk.

Now in its third print run...

FRACTALS EVERYWHERE

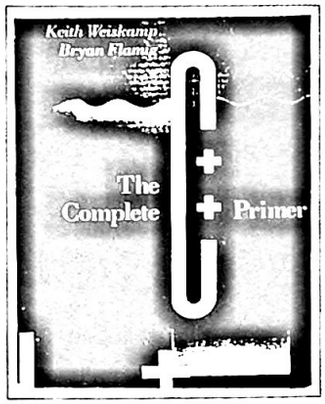
Michael F. Barnsley

"This is a fetching book, with beautiful computer graphics....The material is well chosen and hangs together coherently, and the writing conveys a proper sense of excitement....If you want a genuine understanding of fractals as mathematics, and not just as attractive images, this is an excellent place from which to start."

—NATURE

1988, 424 pages, $39.95 /ISBN: 0-12-079062-9

THE COMPLETE C++ PRIMER



Keith Weiskamp and Bryan Flamig

This is the first comprehensive guidebook and tutorial presenting C++ in a friendly, easy-to-follow format. Including features of the new version 2.0, the book contains numerous programming examples, all of which are designed to work with standard C++ translators and compilers. A wide range of programmers—academic, professional, and technical—will find this tutorial far more readable than existing books on the subject, which are often little more than reference guides. While a basic knowledge of C programming is assumed, all new C++ concepts are introduced with examples that readers are encouraged to try for themselves.

CONTENTS: Introducing C++. Introducing Object–Oriented Programming. Introducing Classes. Classes in Action. Introducing C++ Functions. Putting Functions to Work. Function and Operator Overloading. Working with Constructor and Destructor Function. Inheritance and Class Hierarchies. The C++ Stream I/O System. Record-Oriented File I/O. A Simulation Project Using C++.

In Paperback: $39.95/ISBN: 0-12-742687-6
December 1989, c. 536 pages

INTRODUCTION TO PARALLEL PROGRAMMING

Steven Brawer

This book is the first practical guide to parallel programming written for the applications programmer with no experience in parallel programming and no formal computer science training. You will find all the basic concepts illustrated by examples in a simplified FORTRAN, including the parallel programming model, the creation of multiple processes, memory sharing, scheduling, and data dependencies.

June 1989, 436 pages, $39.95/ISBN: 0-12-128470-0

ACADEMIC PRESS
Harcourt Brace Jovanovich, Publishers
Book Marketing Department #17129, 1250 Sixth Avenue, San Diego, CA 92101
CALL TOLL FREE: 1-800-321-5068
Quote this reference number for *free postage and handling* on your prepaid order ➥ **17129**

Prices subject to change without notice. ©1989 by Academic Press, Inc. All Rights Reserved. CB—17129

Algorithm

THE PERSONAL PROGRAMMING NEWSLETTER

ALGOLETTER 2

Martin Gardner

Four projects for home computer magicians.

EASY PIECES 4

Michael W. Ecker

The Amazing Oracle Machine answers any and all questions.

PERSONAL PROGRAMS 8

Clifford A. Pickover

Chaos and dynamical systems set the stage for strange attractors.

ALGOPUZZLE 12

Dennis Shasha

Dr. Ecco resumes his puzzling career by designing spy codes.

ALGOFACT: Do-It-Yourself Fractals 14

Michael F. Barnsley

Iterated Function Systems make for fantastic image compression and do-it-yourself fractals.

ALGOFICTION: The Donovan Jones 18

A.K. Dewdney

An accountant's brain is taken over by a crew of seven homunculids.

BULLETIN BOARD 24

Software, games and publications available from individuals and small companies.

Algostaff

Editor
A.K. Dewdney

Assistant Editor
John O'Hagan

Managing Editor
Louis D. Magguilli

Copy Editor
Jennifer Connor

Layout & Design
Tammy Golinski

Production Services
Two Oxford West

Address all editorial correspondence to A.K. Dewdney, 2 Oxford Street West, London, Ontario N6H 1P9, CANADA. Address advertising, business and all other correspondence to Louis D. Magguilli, Algorithm, P.O. Box 29237, Westmount Postal Outlet, 785 Wonderland Road S., London, Ontario N6K 1M6. Owned and published bimonthly by 854832 Ontario Inc., doing business as *Algorithm*.

All rights reserved, including the right to accept or reject all editorial and advertising matter. Unsolicited materials must be accompanied by self-addressed return envelopes and postage or dollar equivalent. No part of this newsletter may be reproduced without the written permission of *Algorithm*. Copyright© 1989 by *Algorithm*.

Mail subscriptions: $29.95 (U.S.) per year; single copies $5.00 (U.S.). Overseas add $4.00 (U.S.) for shipping. ISSN 0843-9753

ALGOLETTER

by Martin Gardner

Digital prestidigitation - steps toward a computer magic package

Several years ago I was involved in a software project involving a diskette that would play the role of a magician, performing magic tricks, some of which would mystify even a person running the programs. Nothing came of this project, but in planning material for the diskette (conjuring is my principal hobby) I accumulated an extensive file of tricks that could easily be programmed. Some magicians have thought along similar lines, and programs have been published in magic periodicals, but the field remains wide open to exploration by magicians in collaboration with clever programmers. Here are four typical ideas.

A CARD PREDICTION

The trick begins with the computer throwing on the screen a picture of the back of a playing card. It remains there until the trick's climax.

You are told (in a booklet accompanying the diskette) to shuffle a deck of cards thoroughly, then type into the computer the names of the ten cards, in sequence, that are at the bottom of the shuffled deck. Leave the ten cards in the same position in the deck.

Holding the deck face down, start dealing cards from it face up to form a pile. As you deal, count backward from 10 to 1. All face cards are assigned a value of 10. If a card's value matches a called number, stop dealing and start a second pile. For example, assume you count 10, 9, 8, 7, 6, and a six is dealt on the count of six. Start a second pile, leaving the first pile with the six face up on top. This value obviously was determined by pure chance.

Three more piles are formed in exactly the same way. If no coincidence of card and number occurs, "kill" the pile by putting the next card face down on top of it. If all four piles are killed (this seldom happens), then shuffle the four piles together, replace the cards on top of the undealt cards, and start over.

After the four piles are completed, add the values of the cards on top of the unkilled piles. For example, one pile may have been killed, and a three, queen, and nine are face up on the other three piles. The queen counts as 10, so the sum is 3 +10 + 9 = 22. Count to the 22nd card in the remainder of the deck and note its face.

A computer key, specified in advance (it is always the same key), is now pressed. Slowly the card on the screen rotates to disclose its other side. It will be the very card you seemingly chose at random!

How does the computer know? When you type the names of the ten cards at the bottom of the shuffled deck, into the computer, it ignores all the names except the ninth. This is the card whose face the computer puts on the screen as the face of the rotating card.

A bit of elementary algebra will convince you that the ninth card from the deck's bottom is always the one selected. The trick's most amusing aspect is that spectators have a strong (but completely false) impression that somehow the computer was able to put the chosen card on the screen, its face hidden, before the deck was shuffled.

GUESSING A WORD

In this trick you assist the computer in a sneaky way. The trick is intended for a group of onlookers who do not suspect that you play a role in its working.

Someone freely names a four-letter word. The computer then asks that person a series of four questions. Onlookers watch over your shoulder as you type his or her replies onto the screen.

The program can hold a list of, say, a hundred questions so that the four it asks are different each time the trick is repeated. The questions are totally irrelevant. For example, the first question could be: "In what city and state were you born?"

The next question could be: "What is the first line of one of your favorite poems?" The next two might be: "Who played the male lead in Casablanca?" and "What's the sum of ten thousand and fifteen thousand?" In each case you type exactly what the person tells you.

When a certain key is pressed, the chosen word flashes on the screen.

You must, of course, secretly signal the word. There are many ways to do this. Here is one of the simplest. The computer is programmed not to show on the screen the letter you type after hitting the space bar for the first time. Suppose the chosen word is "pink." When you type the answer to the first question, start the second word with p, then without hesitation go on to the letters of the second word. The screen will not show p or move the cursor. Spectators will be watching the screen, not your fingers. In the same way you give the computer i, n, and k as you type answers to the last three questions.

The first published trick of this type, in which you secretly cue the computer while you type answers, was invented by Chris Morgan, former editor of *Byte*. In Morgan's version, the identity of a chosen card is conveyed by hitting or not

hitting the space bar after you finish typing the answers to yes or no questions. You'll find this excellent trick described by Kee Dewdney in his *Scientific American* column for August 1986, and by Morgan in *Byte*, Fall 1979. Morgan provides a complete program in BASIC.

A HAT TRICK

The screen shows a row of, say, three hats, brims down, and all alike. They are labeled ABC. Pressing a corresponding letter raises a hat a few inches. A second pressing lowers it. Beside the hats is a small bunny.

Stand on the far side of the room, back to the audience, while someone raises a hat, uses the mouse to put the rabbit under it, then lowers the hat over the rabbit. (See the illustration below.) You turn around, return to the computer, and at once press the key that lifts the hat that hides the bunny. How do you know which hat to raise?

In the previous trick you secretly cue the computer. In this one, it secretly cues you. While the hat is being raised, the computer imperceptibly alters the picture of the hat one step to the right. (Assume ABC to be cyclic.) This can be done by placing a tiny dot somewhere on the hat, or by removing a tiny dot, or any other change too small to be noticeable unless you know where to look. On the off chance that a person is closely watching the hat being raised, the change is made on another hat.

The trick obviously has many variants: a row of walnut shells and a pea; a row of inverted cups and a small object such as a ring, marble, or match folder; a row of envelopes and a thousand-dollar bill; a row of doors behind which a leprechaun hides; and so on.

GUESSING A DIGIT

Ask anyone for a dollar bill, and call attention to its eight-digit number. Read these digits aloud slowly. As you do so, ask the bill's owner to think silently of any nonzero digit named.

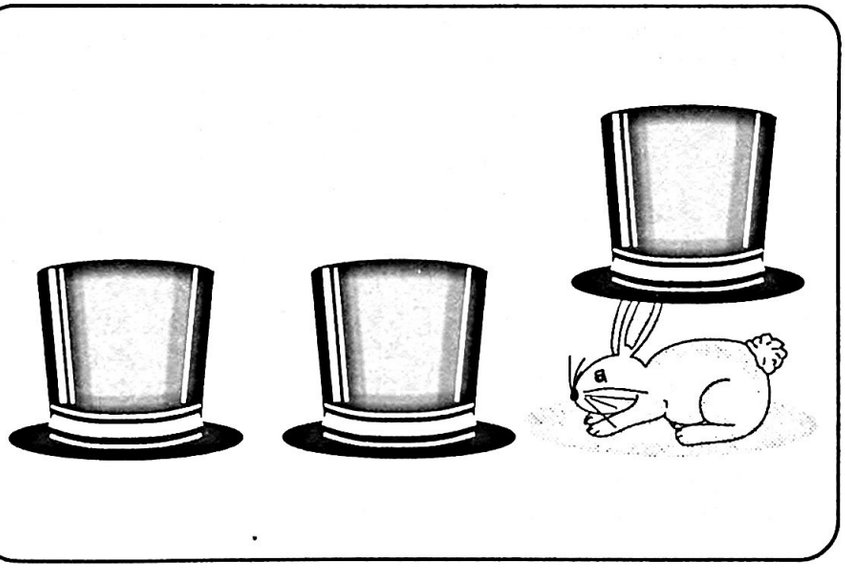
Hand back the bill and tell the person to give the computer, in any order, all the digits except the one mentally selected. The computer screen remains blank. While the owner does this, you stand across the room with your back turned. Thus you have no inkling of what digit he or she thought of.

You now return to the computer and type: "What digit is ___ thinking of?" (including the person's name in the blank space). The chosen digit instantly appears on the screen.

The secret: As you call off the digits at the start of the trick, mentally add them in your head, "casting out nines" as you go along. This is easily done by adding the digits in any two-digit sum and remembering only the single-digit result. For example, if the first number is 9 and the second is 8, you say to yourself 9 plus 8 is 17, and 1 plus 7 is 8. You remember 8. This is known as the digital root of the sequence.

You must secretly tell the computer this number. One way to do it is to press the letter key directly below the digit on the keyboard's top row. As before, you can do this just after the space between the first two words of the question. The computer subtracts from this digital root the digital root of the set of seven numbers earlier given to it by the bill's owner. (The computer has of course determined the digital root of those seven numbers before you send it the digital root of all eight.) If the result is positive, that is the number chosen. If zero or negative, the chosen number is the result plus 9.

I hope these four tricks suggest the flavor and variety of amusing digital magic that computers can perform. There are endless possibilities involving cards, coins, dice, matches, and other common objects, as well as words and numbers. In many cases it is only a question of adapting a well known magic trick to a computer presentation.



Your victim hides the rabbit under a hat.

EASY PIECES

by Michael W. Ecker

The Amazing Oracle Program

Imagine introducing your computer to friends as the Amazing Oracle Machine. "Ask it any question you like," you say; "the question may be mathematical, historical, philosophical." Even computer illiterates will be skeptical: "ANY question?" You may reassure them that even if the Amazing Oracle Machine doesn't know the exact answer, it's prepared to make a pretty good guess.

It sounds impossible, as though artificial intelligence had suddenly caught up with science fiction in one enormous leap. But not only is the feat not impossible, programming your computer to behave with such extraordinary intelligence happens to be an Easy Piece. The program I'm going to show you will convince your friends that you possess the secret that has eluded 40 years of research in artificial intelligence.

Of course, it's a trick, but one that a little practice will make perfect. The effect seems straightforward enough: Your friend asks, addressing the computer directly in spoken English, "What's my dog's name?"

You say, "I'm going to type in a clue if you don't mind."

You type and the following phrase appears on the screen.

WHY GRANDMA -
WHAT BIG TEETH YOU HAVE!

Your friend does not see how a clue like that could possibly lead the computer to respond with "Mike" (the dog's real name). Yet, that is exactly what happens. A moment of thought and the computer responds with MIKE.

Your friend is flabbergasted.

How does the program work? There's a clue buried in the sentence above that starts "You type and the...." But rather than give the whole thing away, I'll change the subject. You will need a little background in loops and string processing before you can understand the Amazing Oracle Program.

LOOPS AND STRINGS

In *Algorithm* 1.1, we learned how to manipulate numeric variables, registers in the machine that store numbers that can change from time to time. The same registers may also be used to store characters (such as alphabetical letters). Therein lies the basis of our trick. But even before we enter the world of characters and strings of characters, we must touch on the repetitive world of loops.

Perhaps one of the simplest examples of a loop is one that prints out some message, say, a hundred times. In BASIC, the language of Easy Pieces, here is one way that such a loop could be programmed:

```
10 K=1
20 IF K <= 100 THEN PRINT
"THIS IS MY MESSAGE"
ELSE END
30 K=K+1: GOTO 20
```

Line 10 initializes K to 1, line 20, which you should be able to enter on one line of your program, performs a certain test, and line 30 adds 1 to K, then directs the computer back to line 20 again. The continuing repetition that results is called the "loop." In line 20, the test concerns the current size of K; as long as K happens to be less than or equal to 100, the computer will print THIS IS MY MESSAGE. But as soon as K exceeds 100, the IF-test fails and the THEN part of the instruction is not executed. Instead, the ELSE is "executed" and the program ENDs. Notice how two instructions may be packed into the same line by separating them with colons. In line 30 the instructions to add 1 to K and to go back to line 20 are placed on the same line. This is very handy for writing more compact-looking programs.

A more natural way to construct a loop is to use the built-in BASIC commands, FOR.....NEXT, as in the next example:

```
10  FOR K=1 TO 100
20  PRINT "THIS IS MY
    MESSAGE"
30  NEXT K
```

In general, the structure of a FOR...NEXT statement can be indicated by a kind of fill-in-the-blanks diagram.

```
FOR_____=_____TO____
_____________________
_____________________
NEXT_____
```

The first blank is filled in with the name of a variable such as K. The next blank is filled with a starting value for this variable and the blank after that (the one after the TO) is filled in with the ending value. The blanks that follow the first line represent lines of program making up the so-called "body" of the loop. I have shown just two such lines but there may be any number. The last blank, the one after NEXT, is filled in with the name of the loop variable, just like the first blank.

When I said, "any number" of lines above, I meant it. Indeed, there might be zero lines. But why repeat nothing 100 times? To waste time, of course. The "moment of thought" which I mentioned that the computer devotes to answering the dog (or any other) question involves just such a time-wasting loop:

```
FOR DELAY = 1 TO 2500
```

Here, I have chosen to call my loop variable by the name DELAY to remind myself what this loop is for. I

could just as easily have used K or any other name.

Would you like to impress your friends with your computer's number-crunching prowess? Enter the following program (using the advice in my last column, if you have it):

```
10 FOR N=1 TO 1000
20 PRINT N, SQR(N)
30 NEXT N
```

When you run this program, you will witness two columns of numbers racing down your screen. The first column will consist of the numbers 1, 2, 3, and so on, up to 1000. Beside each of these numbers, in the second column, you will see its square root printed! Each time the computer executes line 20, it prints N, the loop variable, and SQR(N), the value of its square root worked out in the blink of an eye by the square-root function built into BASIC.

The first character command you will need to know about is called INKEY$. This command tells the computer to wait for a key to be pressed before proceeding. You can change the behavior of the root-finding program above by inserting an extra instruction that uses the INKEY$ command:

```
10 FOR N=1 TO 1000
20 PRINT N, SQR(N)
25 IF INKEY$= " " THEN 25
30 NEXT N
```

The conditional statement in line 25 tests whether a key has been pressed. Before any key is pressed, INKEY$, which also behaves like a variable, is empty. Two consecutive quote marks symbolize this condition. Line 25 says, in effect, "As long as INKEY$ is empty, go back and execute line 25 again. In other words, the computer will wait for you to press some key, any key. Now, instead of racing through the square roots of 1000 numbers, the computer will politely wait for you to tap a key before it proceeds to compute the next square root. Although this small program has only pedagogical interest, the same idea is used by the Amazing Oracle Program.

So much for loops. What, exactly, are strings? Any sequence of alphabetic (or other) characters makes up a string. This very sentence is a string that contains 94 characters, including blanks, commas, and numbers. String variables are just like other variables except they contain strings instead of numerical information. String variables usually have a $-sign appended to them. To find the length of a string such as A$, you may use the LEN function. Thus if A$ happens to equal "hello," then LEN(A$) will equal 5. Note that whenever we define a particular string variable, we enclose the characters that make up the string in quotation marks.

Strings may be concatenated, that is, placed end to end and joined together. If A$ = "Hello " (note the space) and A$ = "Bill", then A$ + B$ = "Hello Bill". Strings may also be taken apart by using the MID$ command. In general, when MID$ is applied to a string, it plucks out all the characters from one point to another and these characters become a new string that MID$ defines. For example, if C$ = "Hello Bill", then MID$ (C$, 4,5,) = "lo Bi". In other words, the new string starts at the fourth character of the string "Hello Bill" and continues on for a total of 5 characters. The first number after the string variable tells MID$ where to begin removing characters and the second number tells how many characters to remove.

In the Amazing Oracle Program, the MID$ command will be used to dig out one character at a time from a previously stored sentence. The characters thus removed may be stored in other string variables; for example,

```
I$ = MID$(A$,N,1)
```

stores the Nth character of the string A$ in the string I$.

Finally, you must know about READ and DATA statements. When your computer executes a READ command such as READ X$, it looks for a DATA statement elsewhere in the program. There it will find a string which it stores in the string variable X$. Thus, if elsewhere the program contained the statement,

```
DATA "ROCK-A-BYE BABY"
```

the statement READ X$ would cause X$ to contain the string "ROCK-A-BYE BABY". Each time your program executes a READ statement, it reads the next available DATA statement. To avoid the possibility of running out of DATA statements, the RESTORE command tells your program to start READing the DATA statements all over again.

HOW THE PROGRAM WORKS

The key idea behind the Amazing Oracle Program is that your victim, in watching you type at your keyboard, assumes that what appears on the screen is what you have just typed. In this case, nothing could be further from the truth! Look, for example, at lines 110, 120, and 130 of the program in the box on page 6. Line 110 reads the strings in the DATA statements in line 190 into the string variable X$ and then sets the variable X (not the same as X$) to be the length of this string. Line 120 uses the INKEY$ command to see whether you have typed a new character yet. As soon as you do, the computer jumps to the next instruction, the one on line 130. Here, it adds the character you just typed to a string called SENTENCE$. On the same line, a counting variable, L, is incremented by one and, finally,

the computer is asked to print the Lth character of the string X$.

What's going on here? As you, the impresario of the Amazing Oracle Program, type one bunch of characters, the computer is printing another. While you type the answer to the question that your victim has asked (or the best guess at the answer), the machine displays one character after another from a string stored in one of the DATA statements.

When you are ready to present the trick, LOAD your program.

As in most magical tricks, presentation is everything. Start by announcing a demonstration of an amazing program you have just written. Naturally, you will call it the Amazing Oracle Program. Tell your intended victim to ask the machine any question out loud; then type RUN. The Amazing Oracle Program will announce its presence and reinforce what you have just said by printing a message on the computer screen (see lines 30 to 80 of the program). The victim tries a question that he or she is reasonably certain your Amazing Oracle Program could not possibly know, say the name of a favorite pet.

"What is the name of my dog?"

Tell your victim that the program will need a clue of some kind in order for its "thought processes" to get started. Make sure your victim can see the screen clearly and invite him or her to watch the screen closely to confirm that the clue you are typing does not give the machine an unfair amount of information. In doing so, you are distracting the victim's attention from the keyboard.

As your friend (or foe) stares at the screen, carefully watching the emerging clue, you type the answer, say, "Mike." Of course, this string is much shorter than any of the clues stored in the DATA statements, so, as soon as you have typed "Mike," pat the space bar about 30 times, rapidly, then press the RETURN key. This will ensure that the entire "hint" is printed by the program — while appearing to have been typed by you. In all events, the computer already knows the answer to the question!

```
10 CLS:CLEAR:PRINT"THE AMAZING ORACLE PROGRAM"
20 PRINT "A breakthrough in artificial intelligence"
30 PRINT:PRINT "I will answer any question in a moment."
40 PRINT:PRINT
50 FOR DELAY=1 TO 2500: NEXT DELAY: REM Time delay loop
60 PRINT "Okay, I'm ready. Ask aloud any question you like."
70 PRINT "My creator will give me a hint and I will reply."
80 PRINT "To retire me, press the BREAK key."
90 PRINT:PRINT:SENTENCE$= " ": COUNT=COUNT + 1:L=0
100 IF COUNT=4 THEN RESTORE: COUNT = 1
110 READ X$: X=LEN(X$)
120 I$ = INKEY$: IF I$=" " THEN 120
130 SENTENCE$=SENTENCE$+I$: L=L+1: PRINT MID$(X$,L,1);
140 IF I$<>CHR$(13) THEN 120
150 PRINT: PRINT "Give me a moment to concentrate..."
160 FOR DELAY=1 TO INT(3000*RND): NEXT DELAY: PRINT
170 PRINT "Aha! I have it!": FOR DL=1 TO 1000: NEXT DL
180 PRINT SENTENCE$:FOR DELAY=1 TO 3000: NEXT:PRINT: GOTO 60
190 DATA "WHY GRANDMA - WHAT BIG TEETH YOU HAVE!      "
200 DATA "THE BETTER TO SEE YOU WITH, MY DEAR.      "
210 DATA "BABY BEAR'S PORRIDGE WAS JUST RIGHT!      "
```

Much remains to explain in the program's operation. How, for example, does it know when you have finished typing in the real answer (such as "mike")? In line 140, the program tests whether the character you have just typed is not equal (i.e., <>) to character number 13, the RETURN or ENTER key. If not, the computer jumps back to execute line 120 all over again. If what you have just typed actually is the RETURN key, however, it goes on to execute line 150 where it prints some chicanery:

"Let me concentrate on that..."

Then it picks a random number between 0 and 1 (done automatically by the RND function), multiplies it by 3000 and then takes the closest integer (INT). The number thus computed becomes the upper limit of a time-wasting loop of the kind I have already discussed earlier in this column. This is a simple way of making your program appear to find some answers easy (a short wait) and others hard (a long wait). Then the computer says,

"Aha...I have it..."

Then it PRINTs the string called SENTENCE, the one the program earlier built up as you typed in the real answer:

"Mike."

The victim looks pale.

"My God, that's amazing! Let me ask another question..."

HOME IMPROVEMENTS

Readers are invited to modify or improve the Amazing Oracle Program in a variety of different ways. As you practise with the program and become sharper at amazing friends and influencing people, you may find yourself dissatisfied with the clues. You may substitute any sentences you like here. Put in something really cryptic, if you like, such as,

"Remember what the old man in the Himalayas told you!"

or even,

"You hardly need a clue for this one, do you?"

Simply take out the sentences currently in the DATA statements and put in new ones in their place. If you decide to have more than three clues, however, you will have to change the COUNT value in line 100. Change the number 4 to a number that is 1 greater than the number of clues you decide to add.

You are also free to change any of the statements made by the Amazing Oracle Machine. These may be found in lines 60, 70, 80, 150 and 170.

EPILOG

Some readers may remember the mid-1980s movie, *War Games*. In it, the protagonist, portrayed by actor Matthew Broderick, does lots of typing at the keyboard. However, the actor was reported as having no typing skills. So, how did he do it? The trick was managed in essentially the reverse way as the Amazing Oracle Program. Instead of inputting answers and outputting clues, the War Games program inputted nonsense and outputted sensible text. By extracting the relevant instructions from our program, readers may construct a program that uses DATA statements containing brilliant passages to demonstrate amazing typing skills. With fingers flying over the keyboard (too fast for friends to see that the wrong keys are being struck), these same passages emerge on the screen with flawless clarity.

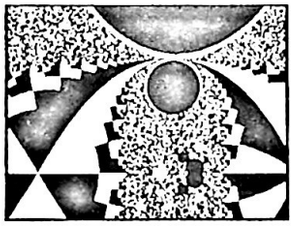
I would enjoy receiving comments, queries and improved programs at the following address:

Dr. Michael W. Ecker
Easy Pieces / Algorithm
129 Carol Drive,
Clarks Summit, PA 18411

Programs may be forwarded in hard copy form but I would prefer them on an MS-DOS diskette, preferably in ASCII format. I can also accommodate TRS-80 tapes and diskettes, Apple II diskettes and TRS-80 Model 100 cassettes. If you expect a reply, please enclose a self-addressed and (if possible) stamped envelope.

Until the next issue, may all your programs prove to be Easy Pieces!

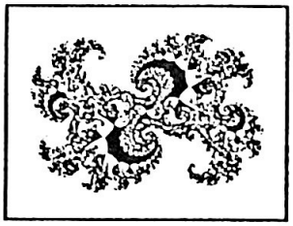
[Dr. Ecker is a professor of Mathematics at Pennsylvania State University.]



ART MATRIX

Free Market System

PO 880 Ithaca, NY 14851-0880 USA
Order (607) 277-0959 24 Hours a day.
Hubbard Dynamical Systems Laboratory



The Free Market System. If you pay the asking price for any product you are guaranteed delivery. Your order will be placed in the Priviledged Queue and filled according to date of arrival. If you submit a bid that is lower than the asking price your bid will be placed in the Free Market Queue and filled according to price and date tempered by our willingness and financial ability to fill it. All bids are queued, including zero. No bid is too low. Free Market bids are not guaranteed delivery as newer higher bids will take precedence over already existing lower bids. If your bid is too low you may never get your product. However the better we do during busy times the lower we are willing to go during quiet times. If you want the product even a little bit, definitely submit a bid. You may be surprised. You can raise or lower your bid at any time without losing your date seniority. ***Give a fractal to a kid today.***

Payment. For orders at the full asking price we are accepting Purchase Orders, Money Orders, personal checks, Mastercard or Visa, Gold, Silver or cash of any currency. Send valuables via registered mail only. For bids in the Free Market Queue we are accepting only UNDATED personal checks or Mastercard/Visa. Your check or card will be cashed when your bid is filled, IF it is filled.

Send for free information about products and our Free Market System.

Nothing But Zooms!, asking $25.00 each plus $3.00 P&H (USA). NBZ! is a 30 minute video of animation and music zooming deep into the heart of the Mandelbrot Set. It contains 12 scenes in living color and one in 3D. This tape was computed on an IBM 3090/600E supercomputer and is 100 percent frame by frame animation of long deep zooms into various points of beauty and interest. Already a classic in the classroom, this tape will make you and your children ask over and over again 'Where did I come from?', 'What am I made of?', and 'What should I be doing with my time?'. Available in VHS and Beta. Also SEC, PAL, U-Matic (extra).

Mini Slide Set (20 slides), asking $20.00 each plus $3.00 P&H (USA). Our mini slide set is a carefully selected cross section of our Professional slide set (155 slides). Many different equations are represented in the full set including quadratics, cubics, Newton's Methods, quintics, rational maps and many more. Each slide has a scientific table in the image containing the name of the image, the name of the program that made the image and its coordinates. These programs are available as fortran source code on our two disk set.

16 by 20 photographic prints, asking $25.00 each plus $3.00 P&H (USA). We have three prints presently in stock, 'Mandelbrot Set', 'Nautilus Shell', and 'Newton's Julia'. The second two may be seen in the June Index issue and the Oct 28 issue of Science News.

T-shirts, asking $15.00 each plus $3.00 P&H (USA). White T-shirts in all sizes with the Nautilus Shell on the front. High quality cotton and accurate reproduction of color. Other designs available. ***Want something we don't offer? Let us know what it is.***

Giant Fractal Poster, asking $15.00 each plus $3.00 P&H (USA). This is a 36 by 24 uncut pack of postcards with 36 different images. Also available as a postcard pack. ***Write for wholesale prices or to get your name on our mailing list.***

PERSONAL PROGRAMS

by Clifford A. Pickover

Close Encounters with Strange Attractors

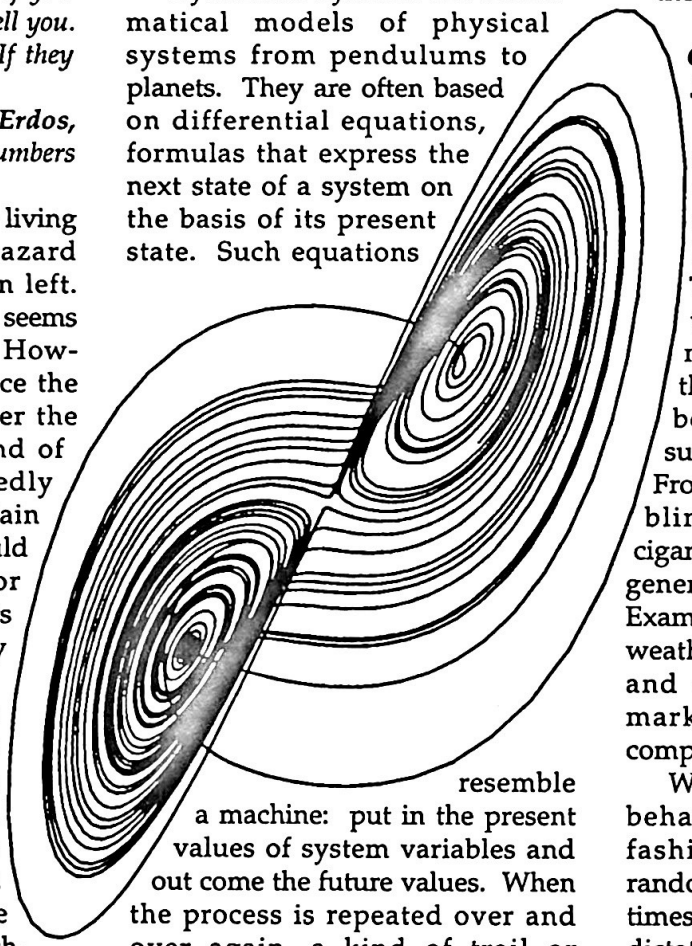
"It's like asking why Beethoven's Ninth Symphony is beautiful. If you don't know why, someone can't tell you. I know numbers are beautiful. If they aren't beautiful, nothing is."

Paul Erdos,
The Man Who Loves Only Numbers

A fly buzzes around your living room in a seemingly haphazard manner. It dives right, then left. Then it does a loop. The path seems random and unpredictable. However, if you were able to trace the path that the fly travels over the space of an hour, some kind of pattern would undoubtedly emerge. For one thing, certain areas of the living room would be favored. The tracery, or graph of the fly's movements might even make a pretty design!

In a similar fashion, certain mathematical equations known as dynamical systems can seem to act haphazardly, yet produce patterns that are both meaningful and pretty. The patterns emerge with the aid of computer graphic techniques to be presented below. Later, I will show you how to create your own mathematical "fly" that follows a dynamical system and, in the process, generates strange attractors, the very image of chaos.

Traditionally, when physicists or mathematicians witnessed complicated phenomena, they assumed complicated underlying causes. In contrast, many of the shapes that emerge in the study of dynamical systems are fantastically complicated, yet their "cause" may be a dynamical system with the simplest imaginable equation! For computer graphics enthusiasts, the payoff is immediate: a simple program that produces images that have both aesthetic and scientific value.



DYNAMICAL SYSTEMS

Dynamical systems are mathematical models of physical systems from pendulums to planets. They are often based on differential equations, formulas that express the next state of a system on the basis of its present state. Such equations resemble a machine: put in the present values of system variables and out come the future values. When the process is repeated over and over again, a kind of trail or trajectory develops. Like the trail of a jet aircraft, the trajectory can be seen—but only with the appropriate use of computer graphics.

The behavior of planetary systems, fluid flow, diffusion of drugs, industrial systems, the vibration of aircraft wings, all this and much more can be described by differential equations. The study of dynamics began in the early 1900s with the work of the French mathematician Poincaré and the Russian Lyapunov. Quiet for the first half of the century, dynamics was revived in the 1960s with the advent of the computer. It became possible for the first time to watch the future trajectory of a dynamical system by plotting its successive states as predicted by a differential equation. But scientists of the 1960s had no idea what lay just around the corner. Chaos.

CHAOS IN DYNAMICAL SYSTEMS

Chaos may lie at the heart of any dynamical system in which there is a very sensitive dependence of the system's behavior on initial conditions. This means that if you change the value of a system parameter or variable, even slightly, the behavior of the system will become very different after a surprisingly short period of time. From chaotic toys with randomly blinking lights to eddies of cigarette smoke, chaotic behavior is generally irregular and disorderly. Examples of chaotic systems include weather patterns, some neurological and cardiac activity, the stock market, and certain types of computer networks.

Whether a dynamical system behaves in a seemingly orderly fashion or whether it jumps randomly about, it is behaving at all times strictly in accordance with the dictates of its underlying equations. Such equations and formulas can be studied, particularly with the aid of computer graphics.

Consider, for example, the famed Lorenz attractor. In 1962, the MIT meteorologist, E.N. Lorenz, attempted to develop a comprehensive mathematical model of weather. Lorenz simplified his differential equations until just three were left:

$dx/dt = 10\,(y - x)$
$dy/dt = 28x - y - xz$
$dz/dt = xy - (8/3)\,z$

The variable t stands for time and the three differentials dx/dt, dy/dt, and dz/dt, give the rate of change with time of the system's three principal variables, x, y, and z.

If we were to plot the behavior

of this system by displaying a point (x, y, z) at each of several closely-spaced values of t, we would see trajectories that trace the outline of what looks like a squashed pretzel, as shown in the figure on the previous page. The surprising thing about Lorenz's system is that if we were to start out with two slightly different initial points, say (.6, .6, .6) and (.6, .6, .6001), the resulting trajectories at first appear to coincide but soon diverge. The actual chaos shows up in the way successive trajectories orbit one lobe of the pretzel or the other, seemingly in random fashion.

There is an underlying pattern. The system is at all times deterministic. It's just very hard to predict what it will do next short of actually simulating it on a computer and watching a graphic display of its behavior. In fact, no matter what state we start the system in, it always ends up tracing out the double pretzel shape. The double pretzel represents the behavior to which the system is attracted, the system's attractor. An attractor in a dynamical system which is capable of chaos often has a strange shape, a fractal.

To create a graphic image of the Lorenz attractor, we need to solve the system of equations given above. To develop such an image, we shall need some numerical techniques that produce accurate values for x, y, and z as functions of time, t. The most straightforward approach I know involves replacing dx by the difference (xnew - x) between the new value of x and the current one. The method replaces the time step dt (normally thought of as infinitesimal) by a definite period of time symbolized by h. This method runs more quickly on a computer than more accurate techniques such as the Runge-Kutta method. To create a picture of the Lorenz attractor in the x-y plane, simply plot pairs of points (x, y), and omit the z-value. An algorithmic outline is displayed in the box below.

Lorenz Attractor Algorithm

```
frac <- 8/3
for i <- 1 to npts
    xnew <- x + 10h(y-x)
    ynew <- y + h(28x-y-xz)
    znew <- z + h(xy-zfrac)
    x <- xnew, y <- ynew, z <- znew
    plot (x, y)
```

(try h=0.01, npts=4000 and starting values x, y, z=.06)

As the environment of a dynamical system changes, the parameters which represent the environment also change and the system may begin to approach chaotic behavior. One sign of the onset of chaos in a dynamical system is the phenomenon of period-doubling. No system shows this phenomenon more clearly than the so-called logistic equation first explored by the physicist, Mitchell Feigenbaum. Once again, computer graphic techniques show the behavior in exquisitely fine detail in the form of a bifurcation plot.

Among the first dynamical systems found to be capable of deterministic chaos was the famed logistic equation:

$$x \leftarrow rx\,(1-x)$$

This nonlinear equation maps a value for x into a new value for x by forming a product of r (the system parameter) with x and (1-x). The equation can be thought of as a series of snap-shots of a dynamical system. Such discrete behavior is typical whenever we simulate a system of differential equations on a computer. For every possible value of the parameter r and initial value x, a sequence of numbers, is generated. A bifurcation diagram of this equation displays the numbers in this sequence as the position of points on a horizontal line. At certain values of the parameter r, the behavior of the sequence is quite orderly. It converges to a single value. However, as r increases, there are suddenly two points to which the sequence is attracted. It flips back and forth between them, getting closer and closer to each. The period of the system's behavior has doubled. As r increases still further, the two-point attractor suddenly becomes a four-point attractor, then eight. These are all "predictable attractors." At other values of r, however, the sequence seems attracted to no definite set of values and takes on apparently random behavior.

We will explore a closely related, little known dynamical system first studied by the mathematical ecologist Robert May in 1976. We can examine May's equation in the form of a bifurcation diagram. In doing so, we will be performing a graphic exploration of all possible behaviors of the equation.

$$x \leftarrow rx\,/(1+x)^b$$

Here, x is the same kind of variable we saw in the logistic equation. This time, to obtain the next value of x the product rx is divided by (1+x) instead of multiplied by (1-x). The divisor is also raised to the power of b, a second parameter connected with May's dynamical system. The upper illustration on page 11 shows how this system behaves as the value of the parameter r gradually increases. For each value of r, the behavior of the system is shown as a sequence of points on an invisible vertical line. As r increases, the vertical line moves to the right, so to speak.

For small values of r on the left of this bifurcation diagram, the behavior is relatively tame. In fact, on the far left the sequence seems to converge to a one-point attractor. As the value of r increases, the ultimate values of the sequence

Bifurcation Plot Algorithm

```
for r <– 0 to 125 in steps of 0.31 /scan r
   x <– 0.9                        /set initial x
   for i <– 1 to n                 /iteration loop
     x <– rx/(1+x)^b
     setcolor (i)                  /assign color
     plot (r, x)
```

(try b=5. The step can be varied to produce finer or coarser resolution.)

system has bifurcated from one to two points. More bifurcations give rise further on the right to chaotic behavior. The sequence never settles down to a simple pattern of attraction. Indeed, certain values of r might be used to turn May's equation into a reasonably good random number generator.

The richness of this bifurcation diagram contrasts sharply with the simplicity of the generating formula we have used. Apart from their curious mathematical (not to mention aesthetic) appeal, bifurcation maps of chaotic dynamical systems can lead to important insights into such vital systems as the human heart (some forms of heart failure appear to be due to arrhythmias that develop as a result of chaotic behavior in the cardiac nerve plexus), weather systems, neural networks, noisy electronic components, and even population biology. For example, apparently erratic fluctuations in animal populations can result from a purely deterministic relationship between predator and prey. As May has pointed out, the fact that simple deterministic equations, such as those presented here, can produce trajectories that look like random noise has disturbing practical implications.

For readers who would like to write a bifurcation diagram for May's equation, I have included an algorithm on which such a program can be based. The algorithm, shown in the box above, suggests a color assignment based on the number of iterations needed to produce a particular point of the plot. If color options are not available, the "setcolor" instruction can be omitted.

A COMPUTER FLY

Readers may recall the fly that I mentioned at the beginning of this article. The trajectory of some dynamical systems resembles the path a fly takes as it buzzes around a room. More specifically, the trajectory can almost repeat itself, never quite returning to the same place. Like a leaf in the turbulent eddy of a stream, the trajectory becomes impossible to predict from an initial position.

What does such a trajectory look like? The lower illustration on the next page was created by a dynamical system that I call a computer fly. Its motion is governed by three simple trigonometric functions of x, y, and z. The equations appear in the algorithmic outline shown in the box on the right.

An initial point (0, 0, 0) is selected and the equations are iterated 5 million times. Each pixel in the illustration has a density that reflects the number of times the fly crossed it while buzzing about an imaginary three-dimensional box. Since the screen is two dimensional, it might be more accurate to say that the density reflects the behavior of the fly's shadow.

If the dynamical system that produced this plot were totally random, the plot would be nothing more than a diffuse scattering of points inside the box. On the other hand, if the systems were absolutely periodic, like a sine wave, the figure would consist of a thin curve in three-dimensional space. As it stands, the figure is delicately poised somewhere between these two extremes. The starting point makes little difference: this shows that the figure is an attractor for the fly. As long as it starts out somewhere near the attractor, it quickly converges to it.

The computer fly algorithm uses a two-dimensional array called *p* to represent the density of the fly's path to each pixel. The density depends on how many times the fly's shadow visits the pixel in question. The resulting array can be plotted directly if you have a computer that employs grey levels on its screen (or printer). If not, you can simply plot a point at (x, y) if p(x,y)>0. Not shown in the algorithm is the conversion of the point (x, y) into a point on your screen. This can be effected with nothing more complicated than high-school mathematics. For example, if the screen is, say, 200 pixels wide and the x-values lie within the range from -2 to +2, then the x-coordinates on the screen might be computed by the formula 50(x+2).

The algorithm has five parameters a, b, c, d, and e. Suggested settings for these are included in the algorithm. Readers who program the computer fly might be

Computer Fly Algorithm

```
a <– 2.24, b <– 0.43, c <– -0.65
d <– -2, e <– 1
initialize array p to 0
initialize x, y, and z to 0
for j <– 1 to 1000
      for i <– 1 to 5000
      x1 <– sin (ay) - zcos (bx)
      y1 <– zsin (cx) - cos (dy)
      z1 <– esin (x)
      x <– x1, y <–y1, z<– z1
      if x1 <2 and x1 > -2 and y1 <2
      and y1 > -2 then
            x2 <– (x1 + 2) width
            y2 <– (y1 + 2) height
            p(x2, y2) <– p(x2, y2) +1
```

The sides of the box within which the fly moves can be changed from -2 and +2 to larger or smaller values for the purpose of expanding or magnifying the picture. The variables width and height are simply the width and height of the display screen in pixels. The iteration limits may also be changed to produce a denser or lighter image.

particularly interested in experimenting with different values of e, which controls the degree of apparent randomness in the system.

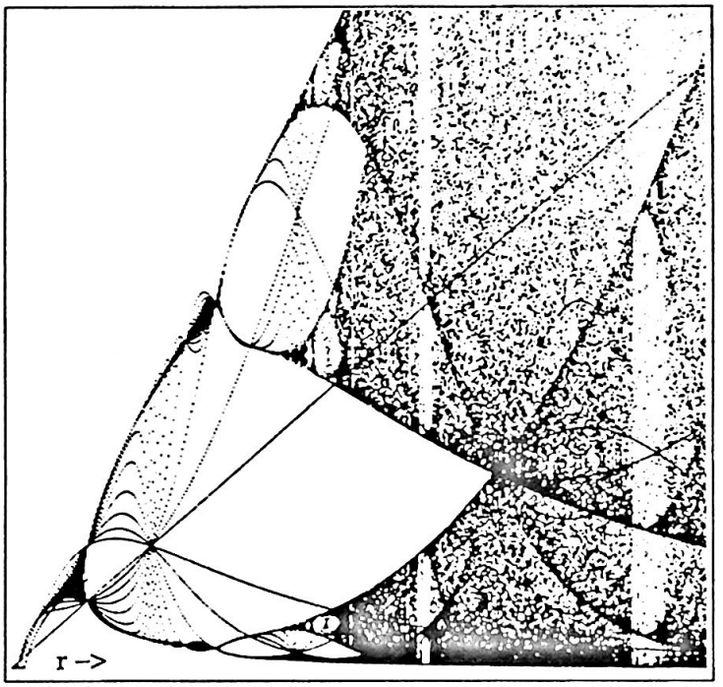
Watch the computer fly as it paints a chaotic figure on the screen or printer. Can you see any long-term regularity in its path? Imagine the plight of a scientist looking for regularity in weather systems. Some scientists would argue that weather itself follows the chaotic dynamics of a strange attractor. Even though it may never be possible to predict precisely phenomena like the weather or the stock market, one might at least foresee overall patterns in their behavior. Its just a matter of mapping the strange attractor itself.

Readers with comments or queries about this article may write to Dr. Pickover at Algorithm / Personal Programs, 37 Yorkshire Lane, Yorktown Heights, NY 10598.

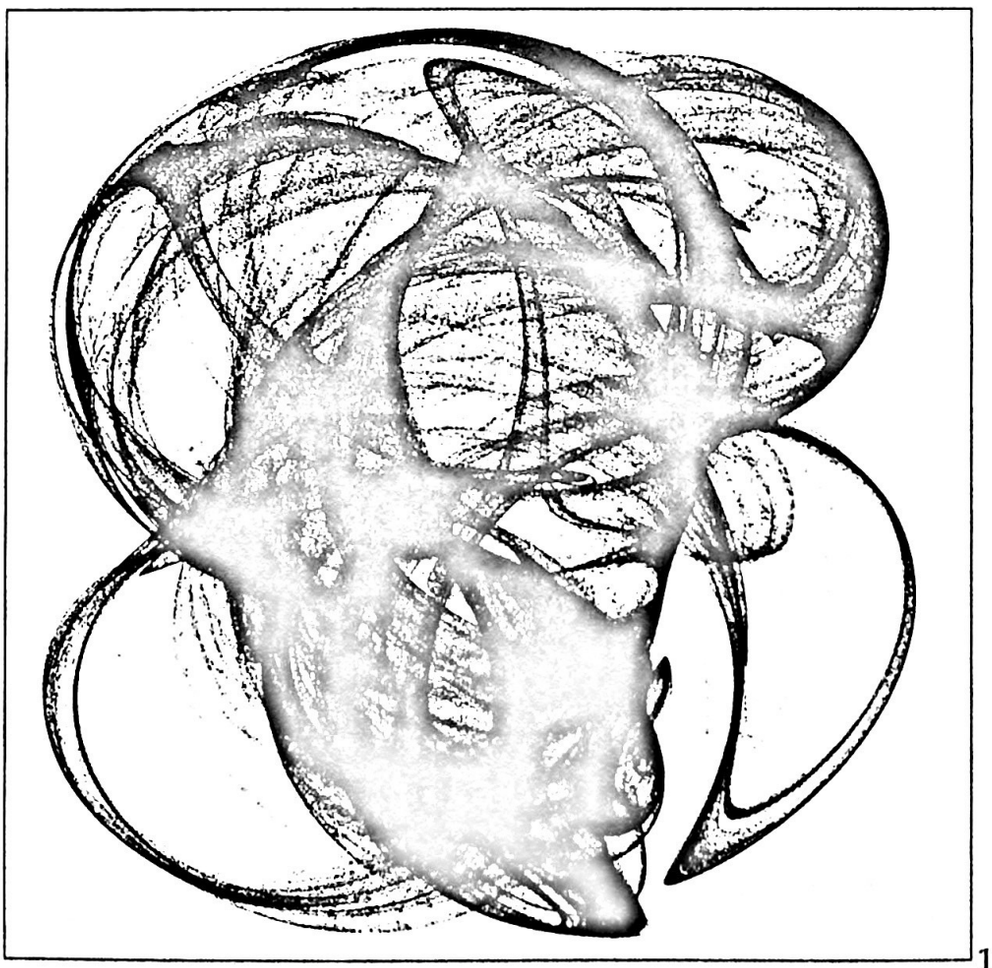
Further Reading:
James Gleick,
Chaos: Making a New Science,
New York, Viking Books, 1987.

Ian Stewart,
Does God Play Dice?
(The Mathematics of Chaos),
New York, Basil Blackwell, 1989.

Paul Hoffman,
"The Man Who Loves Only Numbers,"
The Atlantic Magazine, Nov. 1987



The bifurcation diagram for May's dynamical system.



Activity of the computer fly. ⟶

ALGOPUZZLE

by Dennis Shasha

Spy Codes - The continuing (and puzzling) adventures of Dr. Ecco

In this issue Algorithm introduces Dennis Shasha, about whom more details are given at the end of this column. Shasha's favorite character, the amazing Dr. Ecco, resumes his career (see the bibliography) with this excursion in international espionage.

It was raining heavily as Dr. Ecco, world renowned omniheurist, made his way up the granite steps of the Decentralized Intelligence Agency or the DIA as it is known worldwide. Ecco had been fighting a slight cold and he feared his mental faculties were not everything they might be. What could the director possibly want?

Before he knew it, Ecco found himself standing before the Director of the DIA, a very large man with a gravelly voice:

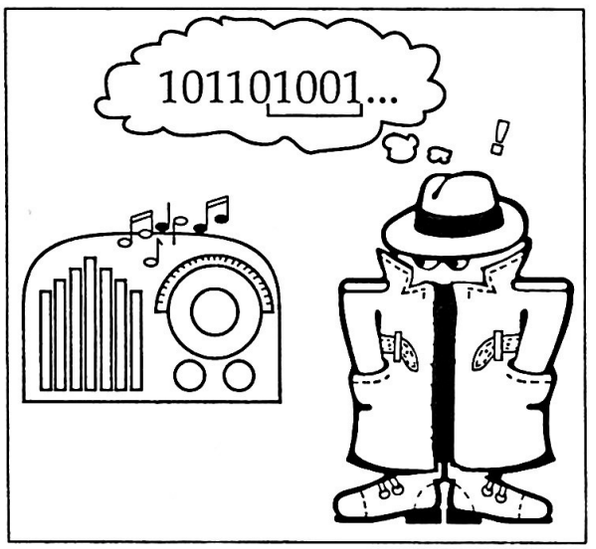
"Everyone denies it but periodically the Voice of America sends messages to our operatives in the Soviet Union. One of our problems is to keep these messages short so the Russkies don't figure them out."

The Director took a puff on his cigar, clouding the air with noxious fumes. Then he continued.

"Here's how the system works. Each agent must receive a message consisting of just four bits. Each bit is encoded in the musical phrase of a song. For example, a certain four-beat sequence of notes means 0 and any other sequence means 1. The problem is that we have nine agents and they all need different messages. As I said, we don't want the song to go on too long or the Soviets may get suspicious. What we do is broadcast one song of no more than 14 phrases in length. The song is structured so that if each agent starts listening at the right time, he will get his message.

"For example, if there were two sequences, 1011 and 0110, we could encode them together into the sequence 011011 and tell one agent to start listening at the beginning and the other agent to start listening at position 3."

Ecco wiped his feverish brow and sighed.



A spy gets his message.

"But how will each spy know when to begin listening?" he demanded impatiently.

"Oh, that's simple," declared the Director, "We encode that in General Electric's advertisement in Pravda."

The answer still seemed incomplete to Ecco. How could they encode such information without encoding the message itself?

But he refrained from asking. The Director laid out a sheet of paper in front of Ecco. It bore the following messages:

0000, 0011, 0100, 0110, 0111, 1011, 1100, 1101, 1110

"What we need, Dr. Ecco, is a sequence of just 14 bits that contains all of the four-bit messages on this piece of paper. I've got a debriefing to attend. Feel free to use my desk. There's brandy and cigars in the cabinet over there."

In spite of his bad cold, it did not take Ecco long to find a fourteen-bit sequence that held all nine messages. (Readers should not be ashamed to take longer than even an ailing Dr. Ecco.) The indisposed omniheurist spent the rest of this time eyeing the liquor cabinet, sternly holding himself from temptation.

Suddenly the Director returned.

"Damn rookie agents! Say, I just talked to our Soviet network operator. We'll have to modify the problem I just gave you. We need to send messages to five more agents. That would make, let's see, 14 messages in total. The problem is that right now we don't know what the messages will be. Can you find a sequence of 19 bits or less? We don't dare make the song any longer!"

The room swam a bit in front of the amazing omniheurist.

"You mean the first nine messages remain the same but you can't tell me what any of the five additional messages will be, and you want me to give you a sequence of nineteen bits or less that will contain all 14 four-bit messages?"

"Precisely," coughed the Director, "One of our own experts just told me what he calls a 'naive upper bound' on the number of bits you will need. He says if you found a 14-bit solution to the earlier problem then there are only five more four-bit messages possible in any case. These would require a maximum of 5x4 or 20 more bits. But we can't afford 34 bits."

"Well," said Ecco, doing a little calculation in his head, "The first list uses nine of the 16 possible four-bit messages. That leaves seven other

possible messages to fit in. There are 21 ways to choose five messages from among seven but I wouldn't want to try all possibilities. In any case, my intuition tells me there's a 19-bit solution."

The Director grunted, "I hope your intuition is correct, Ecco."

But later the Director managed to smile, according to Ecco, when he told me about his adventure after solving the second problem. Would it violate national security if I gave both problems to readers of *Algorithm*? Ecco said it was fine with him.

Answers to both puzzles will appear in the next Algopuzzle column. In the meantime, solutions to the buttonboard puzzle have been passed along to me. Sydney Afriat of Ottawa, Ontario has developed a program that enables one to play with a simulated buttonboard. Press the key for a given button and its four neighbors all change state, from lit to unlit or vice-versa. This is certainly easier and cheaper than building an actual buttonboard from electronic parts.

Bob Trent of Hardinsburg, Kentucky has gone a step further. His program systematically grinds out solutions for an nxn buttonboard. Here is what Trent has found so far: when n = 1, 2, 3, 6, 7, or 8, there is only one solution, symmetric in each case. For n = 4 there are five essentially different solutions. "The obvious conjecture fails," says Trent. When n = 9, there are "scads" of solutions.

Dennis Shasha is an assistant professor of computer science at New York University, and author of the Puzzling Adventures of Dr. Ecco (see below), the notorious mathematical sleuth. Puzzle-solvers and other readers may correspond with Dr. Shasha at the Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012. Enclose a stamped (if possible), self-addressed envelope.

Further Reading

Dennis Shasha,
The Puzzling Adventures of Dr. Ecco,
New York, W. H. Freeman, 1988.

WHAT IS
"FOCUSED ADVERTISING"
?

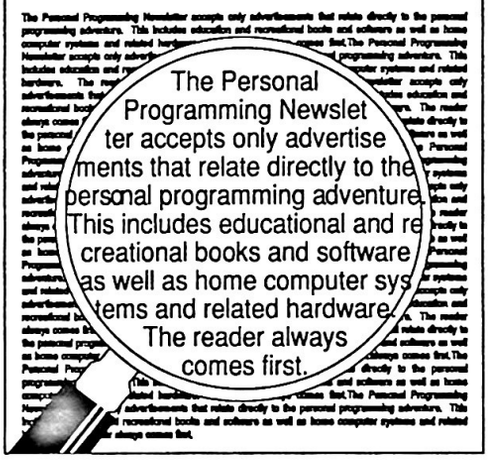


IMAGE CREDITS

Cover: Michael F. Barnsley
Dept. of Mathematics
Georgia Institute of Technology
Atlanta, GA 30332

Personal Programs: Clifford A. Pickover
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598

Algofact: Michael F. Barnsley

Algofiction: A.K. Dewdney

Other Images: Tammy Golinski

The Bulletin Board

The Bulletin Board is an issue-by-issue directory of products and services provided for our readers by individuals and small companies or organizations. We will accept ads for software of educational or recreational value, pamphlets, books, videotapes, clubs, organizations and so on.

To place a Bulletin Board item, send camera-ready copy in the proportions of the rectangle on the right-or fill out by hand and we will set it in type. Be sure to include all relevant information such as address and conditions of distribution.

First placements are free but subsequent placements of an ad (or its equivalent) will be charged at the rate of $30.00 (U.S.) per issue.

Please mail the copy to:
Algorithm P.O. Box 29237, Westmount Postal Outlet
785 Wonderland Road South
London, Ontario, CANADA
N6K 1M6

ALGOFACT

by Michael F. Barnsley

A new image compression technique saves space and spawns fractals.

Traditional computer graphics systems can easily save and redisplay simple geometrical shapes like rectangles, triangles, and circles, with a minimal amount of storage. They simply store a few reference points and apply an equation to the points to fill in the image.

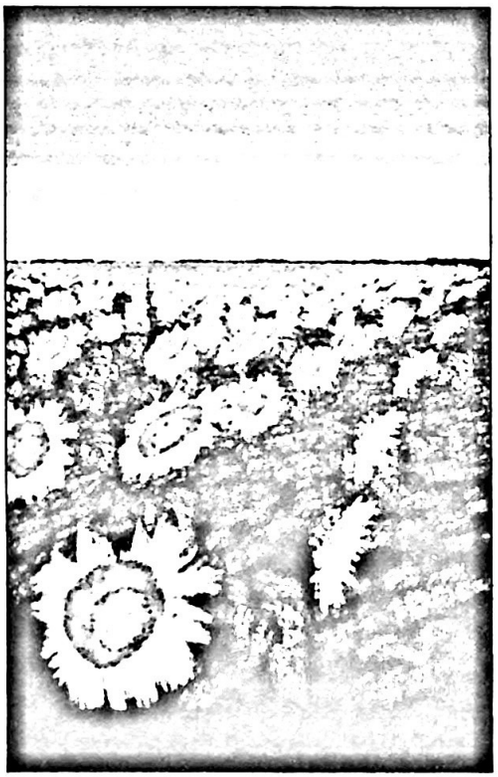
Even simple three-dimensional shapes and color images can be stored and redisplayed with relative ease. They require a little more storage for additional coordinates and color data as well as slightly more complex equations to reconstruct the images.

Natural images that have complex shapes, textures, and subtle color shifts are more cumbersome to reproduce. Forests, seascapes, and mountain ranges lack any kind of pattern that can be represented by the traditional mathematical equations. The standard approach to digitizing such images is to break them down into an arbitrary number of pixels, and store the attributes of each in an array large enough to contain every pixel in the image.

With this technique, even an eight-by-ten-inch black and white photograph, resolved to 300 by 300 pixels per square inch with eight significant bits per pixel, requires almost eight megabytes of memory, too much for most personal computers to handle. Even standard compression techniques that achieve ratios of up to 10 to 1 can only reduce the requirement to just under one megabyte. This still exceeds the capacity of most personal computers.

A new method of image compression is being developed that allows complex real-world images to be represented with only a few thousand bytes of memory.

The novel method uses fractal geometry, developed by Benoit B. Mandelbrot, to represent apparently complex images in terms of the underlying patterns of shape and shading. Fractals have already been used to produce simulated images from nature, like clouds, mountains, and leaves, but until now there has been no way to take an actual image and reproduce it to a reasonable level of accuracy using fractal mathematics.



An IFS-encoded field of sunflowers

The figure above shows an image reproduced from a real photograph that was stored using this method. The storage requirement for this image is approximately 1000 bytes.

OVERVIEW OF FRACTAL IMAGING

There are four basic steps involved in storing and redisplaying an image using fractals. First, the image is digitized using image-processing techniques that detect changes in color, density, and texture to break the image into segments containing visually distinct patterns. A segment might consist of a tree, a leaf, or the sky. It can also be a more complex collection of pixels, like whisps of clouds or a rocky landscape. Some of these image-processing techniques are used to "colorize" black-and-white motion pictures.

Then, each segment is matched with an entry in a library of fractals, a collection of relatively compact sets of numbers called Iterated Function System (IFS) codes that represent the corresponding fractals. The library is organized in a way that allows automated search routines to locate a fractal that approximates the given segment. A mathematical result known as The Collage Theorem guarantees that a suitable set of IFS codes will always be found.

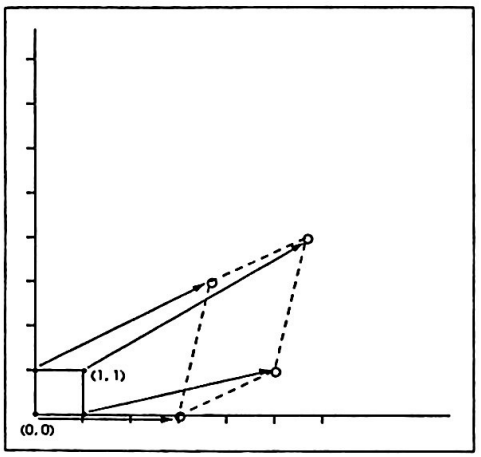
The entire image is then stored as a series of IFS codes and attributes indicating the location, size, and orientation of each segment. To redisplay the image, an algorithm is applied to each set of IFS codes to generate a composite image.

The IFS code concept is what makes this method of image reproduction so powerful. It allows patterns often found in nature to be represented by only small sets of numbers.

CREATING IFS CODES

IFS theory uses affine transformations to express relations between an image and parts of that image. It describes a natural image with the same degree of accuracy that an architect describes a house.

Affine transformations are functions that can change the size, shape, orientation, and position of a



An affine tranformation moves four points.

set of points plotted on co-ordinate axes in n-dimensional space. In two dimensions, using x and y axes only, an example of an affine transformation is:

$W(x, y) = (2x + (1/2)y + 3, x + 3y + 0)$

This can be written in matrix form as:

$$W\begin{bmatrix}x\\y\end{bmatrix} = \begin{bmatrix}2 & .5\\1 & 3\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix} + \begin{bmatrix}3\\0\end{bmatrix}$$

This transformation would move the point (0,0) to (3,0), and point (1,1) to (5 1/2, 4), as shown in the figure above.

A contractive affine transformation is one that moves points closer together. When applied to a given figure, a smaller figure is generated that may or may not overlap the original. The same transformation applied to the smaller figure will produce a still smaller one that bears the same geometric relationship to the second figure as the second to the first, as illustrated in the figure on the right. The fractal potential of such transformations is almost immediately obvious. Here is a way to generate parts that resemble the whole!

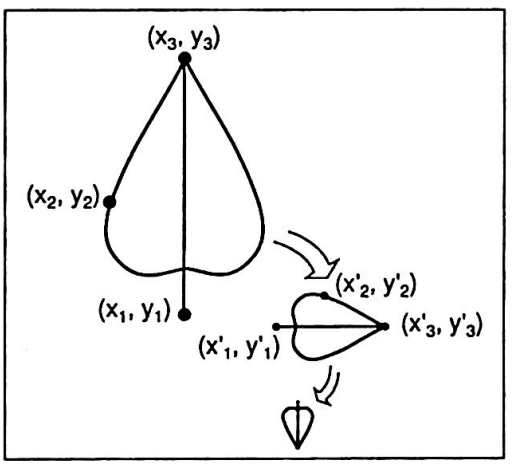
The general form of an affine transformation in two dimensions is:

$$W\begin{bmatrix}x\\y\end{bmatrix} = \begin{bmatrix}a & b\\c & d\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix} + \begin{bmatrix}e\\f\end{bmatrix}$$

where the coefficients a through f are real numbers.

To calculate the values a through f for a given transformation, begin by marking three points on the original image and the corresponding points on the transformed image. (We've designated the points in the first two images of the figure as (x_1,y_1), (x_2,y_2), (x_3,y_3), and (x'_1,y'_1), $(x'_2\ y'_2)$, $(x'_3,\ y'_3)$. Then determine the values for coefficients a, b, and e by solving three linear equations:

$$x_1a + y_1b + e = x'_1,$$
$$x_2a + y_2b + e = x'_2,$$
$$x_3a + y_3b + e = x'_3,$$



A contractive affine transformation moves points closer together

Then find c, d, and f, in similar fashion from these equations:

$$x_1c + y_1d + f = y'_1,$$
$$x_2c + y_2d + f = y'_2,$$
$$x_3c + y_3d + f = y'_3.$$

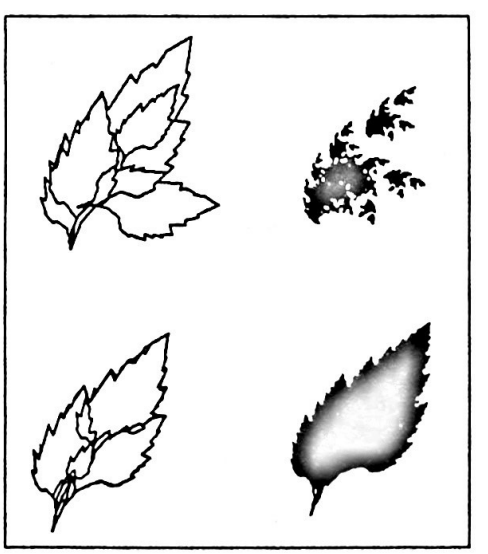
IFS codes are simply sets of such coefficients describing any number of transformations, with an indicator of the"importance"of each transformation in producing the desired image. The indicator is expressed as a probability, p, and governs the relative frequency that each transformation will be applied during reconstruction. For example, the IFS codes for a solid black square are:

W	a	b	c	d	e	f	p
1	.5	0	0	.5	0	0	.25
2	.5	0	0	.5	.5	0	.25
3	.5	0	0	.5	0	.5	.25
4	.5	0	0	.5	.5	.5	.25

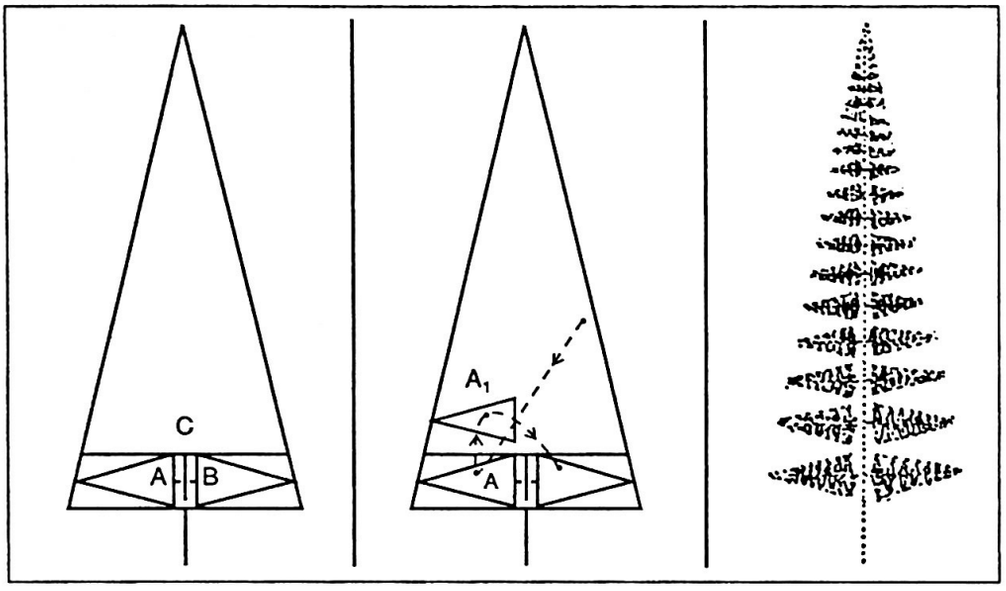
If you take any point (x,y), apply any of the four transformations, and continue applying any of the transformations to each resulting point, you generate a square. (The first few iterations may generate points that do not fail on the boundary of the square. These iterations are not used in producing the image, as shown in the Random Iteration Algorithm in the box on page 16.) Note that the transformations must be used in equal proportion, because the probability of each of the four is 25%.

The IFS codes for any figure can be determined by finding a set of contractive affine transformations that combine to approximate the original image. In general, the more the transformations contract the original image, and the more accurately the transformed images cover the original, the closer the reconstructed image will approximate the original. For more information on determining the best transformations for various images, readers may consult the book *Fractals Everywhere*. (See Further Reading.)

The illustration below shows the importance of choosing good transformations. The image in the upper left was transformed into



Improper and proper collages.



Three transformations of an image-generated collage that nearly covers the originals.

A point is translated by these transformations to corresponding locations in the sub image.

As more and more points are plotted, the resulting image begins to resemble the original.

four contractive images that do not combine well to cover the original image, and when the Random Iteration Algorithm is applied to these transformations, the reconstructed image is a far cry from the original. The lower half of the figure shows the result of choosing good transformations.

RECONSTRUCTING IFS-ENCODED IMAGES

A Random Iteration Algorithm is used to reconstruct an IFS-encoded image. The algorithm in the box on the right generates values for x and y from an IFS code with m transformations:

The algorithm first initializes x and y to 0. Then, it generates 2,500 new versions of x and y by transforming and re-transforming the point. The transformation used at each step is selected at random, based on the individual probability associated with each transformation. The lower the probability, the less often that transformation will be applied. Note that no points are plotted for the first 10 iterations. This gives time for the points to "settle" within the range of the image.

The illustration above shows how a Random Iteration Algorithm can generate an image through affine transformations. In this case the image is a fern leaf. First, a collage of the original image is created, consisting of three smaller versions that approximately cover the original. These were generated with contractive affine transformations A, B, and C. Transformation A transforms the entire leaf into a leaflet growing from the left side of the stalk at the base. Transformation B does the same thing to produce a leaflet on the other side. Transformation C takes the whole leaf into a slightly smaller version that has been shifted along the stalk approximately the distance of one leaflet.

To regenerate the image, the transformations are applied repeatedly to a point (x,y) at a frequency relative to the probability assigned to each. When transformation A is selected, a point anywhere in the big leaf is mapped into the corresponding point of leaf A. When transformation C is applied to this new point, it is mapped into the corresponding point of leaf A_1. (Leaf A_1 is located within leaf C at the position where leaf A would be if leaf C were the original image.) As this process continues, more and more points are filled in, eventually generating an image of a leaf comprised of many smaller images of itself.

The probability you assign to each transformation does not generally affect the characteristics of the resulting image, but it does affect the rate at which various regions or attributes of the image are filled in. The amount of time the algorithm should spend on a given transformation should be approximately equal to the area of the image used to derive the transformation divided by the total area of the image.

The areas are easy to calculate. The area used by the kth transformation is just its determinant, $a_k d_k - b_k c_k$. The area of the image is approximately the same as the sum of the areas associated with all the transformations, in other words, the sum of their determinants.

A value of 0 should be replaced with a very small positive value, such as 0.01, and the other probabilities correspondingly adjusted to keep the sum of all the probabilities equal to 1.

CURRENT DEVELOPMENTS IN FRACTAL IMAGING

The IFS compression technique is computation intensive in both the encoding and decoding phases.

Random Iteration Algorithm

```
1. x <-- 0, y <-- 0
2. for n <-- 1 to 2500
       k <-- random (1, ..., m)
       (x, y) <-- kth transformation (x, y)
       if n > 10, plot (x, y)
```

Approximately 2500 points are mapped, each by a randomly selected affine transformation, into a composite fractal image.

Iterated Systems Corporation of Norcross, Georgia was founded in the spring of 1987 to develop commercial applications of IFS image compression. It has recently developed a video modem that can take a digital image file and reduce it to a proprietary form of fractal compression code in fewer than three seconds. At the receiving end, the images can be reconstructed at the rate of 30 frames per second.

The company has also devised a method of reducing the memory requirements of a full-color image of 512 pixels square, with 24 significant bits per pixel, to fewer than 6,000 bytes.

As computation requirements are reduced further, and compression ratios become higher, full-color animation at video rates over phone lines could become a reality.

The company is also marketing through Academic Press an educational system for learning about IFS theory. The Desktop Fractal Design System runs on an IBM Personal Computer with 640K and works with both EGA and VGA graphics.

Another potential application of IFS imaging is automated image recognition. Since this involves combinatorial searching, and search times increase factorially with the size of the image file, highly compressed images will allow computers to detect subtle differences in images at faster rates. It will be possible for a computer to determine quickly whether an image is of a cumulus or a cumulo-nimbus cloud, for example.

Such advances in computer imaging techniques could have tremendous implications in industry, commerce, and medicine.

Michael F. Barnsley is a professor of mathematics at the Georgia Institute of Technology (Atlanta, GA 30332) and the executive vice-president at Iterated Systems Inc. (5550-A Peachtree Parkway, Norcross, GA 30092).

Further Readings

Michael F. Barnsley,
Fractals Everywhere,
New York,
Academic Press, 1988.

Benoit B. Mandelbrot,
The Fractal Geometry of Nature,
New York,
W.H. Freeman, 1983.

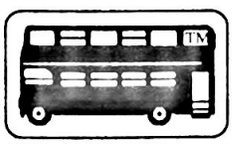
EXPLOR CHAOS

with

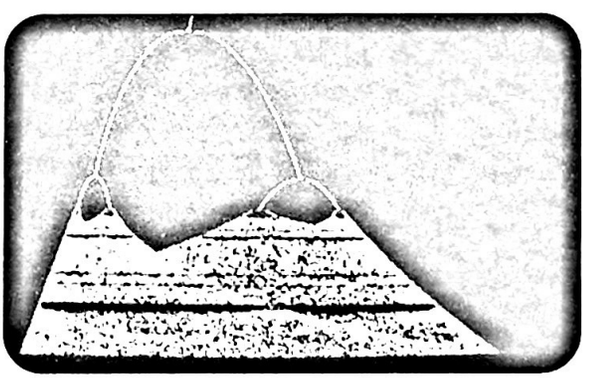
EXPLOR-I

from

TURING OMNIBUS



P.O. Box 1456,
London, Ontario
CANADA N6A 5MZ



For PC and compatible computers
with CGA compatible graphics.

EXPLOR-I is a package of programs designed for amateur and professional explorers alike. Select up to eight programs for your own personal package: Julia and Mandelbrot sets, Wallpaper for the Mind, Exploring Chaos, The Invisible Professor, The Cyclic Space Cellular Automaton, Star Clusters Simulation, Computer Popcorn, Biomorphs. In addition, we now offer the Core War Coliseum, the official Core War programming and tournament package. Write for our free catalogue to-day.

ALGOFICTION

by A. K. Dewdney

Is someone you know a human spaceship?

THE DONOVAN JONES

Behind the appearance we call reality lurks a second level of being, a parallel universe that supports our own. It is a universe not of matter, but of information, where intelligence manifests in forms of its own choosing. One such form is the homunculid.

A homunculid cannot survive in our level of reality unless it finds the proper substrate, a pattern of energy that will contain it in the way a computer contains a program. A homunculid may live for a time within a thundercloud, certain ore deposits, even intricate eddies in a stream. But in these it is cramped. In a human brain it can move and have its being.

In their native level, homunculids range freely over a universe of choice. They make what they need by a process similar to imagination. Common objects are in the common mind.

So it was that seven homunculids were once travelling in the shared reality known as a logic cruiser toward a place called Modus Ponens. Without warning, an axiom storm erupted and the cruiser ran directly into a contradiction. The homunculids braced themselves, as the cruiser faded all about them. They were bumped, as it were, completely out of their own level and into our own. The cruiser appeared momentarily on our level where it manifested, or tried to manifest, as the car being driven by Donovan Jones, a partner in the accounting firm of Melnick, Jones and Leathorne. Jones was on his way home from a particularly trying day at the office.

Unfortunately, the logic cruiser could not manifest as a car and being illogical, it vanished. The homunculids would also have vanished were it not for the substrate they sensed nearby. In a panic, they leapt the gap of nonexistence into the brain of the accountant.

The mind of Donovan Jones disappeared at precisely at this moment. In place of his "I" there now dwelt a "We", seven castaways who groped for each other in the new substrate. Not surprisingly, Jones' car left the road and struck a telephone pole.

The homunculids found each other in the darkness. and spun a common reality, as if the accountant were dreaming. With an electric rustle followed by a popping noise they found themselves standing in a control room like the one in their now vanished logic cruiser. It was a gesture of hope tinged with irony. What, if anything, did they control? They were marooned:

It was Sorebrum the navigator who stepped up to the master control panel where Brunt the pilot sat staring at the unfamiliar controls. It was Sorebrum who discovered the on-off switch.

Donimunt the Captain stared moodily at the display screen in front of the control room. Since it had first come to life with the giants in white coats, he had lived in a state of fear and uncertainty. He had seen many strange sights in his travels but none stranger than the beings on the screen. Their bodies were enormous and badly out of proportion. Their jaws and hands were too big, their eyes and ears too small. Their speech, as it came to Donimunt through two loudspeakers on either side of the control room, consisted of bass growls that spoke solemnly of paralysis, x-rays and trauma.

He had tried many times to will the giants away, to imagine something in their place. But the giants did not change. He had ordered the whole crew to join in the effort but nothing had happened. The giants peered and scowled, came and went.

He frowned. If this was a ship, what was its nature? He longed to be off, away from these monstrous beings and the great, boxlike cavern that enclosed them. He swiveled his command chair to fix Sorebrum in his shining black eyes.

"Navigator Sorebrum. Report our situation."

"Captain, I am frankly astonished that our screen portrays what appears to be an independent reality. It's nothing less than a miracle. But now I must ask..."

"Si-tu-a-tion Re-port!" Donimunt gritted his teeth.

Sorebrum stroked his tiny goatee furiously.

"Precisely, yes, as I was about to say, an independent screen image implies a ship, but exactly what kind is not yet clear."

"Come, come. I don't have all day."

"There are clues, of course. I dare say those giants are working on our vessel at this very moment."

This intelligence brought a glint to Donimunt's eyes. It made sense. Drydock. Refitting. Perhaps they could castoff soon.

At that moment, one of the giants, an enormous clumsy-looking female by all appearances, reached behind the control room. She pulled on something. The image on the screen abruptly shifted to reveal a giant body extending before them, covered by a vast blanket of woven fibers.

One of the giants seemed to look directly into the control room. It said:

"Donovan Jones. Can you hear me? Can you move your hands or feet? Can you blink your eyes"?

"Aha," cried Sorebrum. "I think I have it!"

"He thinks he has it"! cried Fuzbuji, the soical director.

It was Fuzbuji's job, among other things, to keep the crew in a jolly mood. This was his first opportunity to display cheer since the axiom storm and the dreadful shipwreck.

"Shut up and let the navigator speak quietly in my ear!"

"It's perfectly obvious," whispered Sorebrum, "That giant body is our ship! Merely look down your own body. Our screen shows the view from the eyes of that body and the ..."

"That's enough, navigator. Now hear this! I, Donimunt, have decided that our ship is a giant body like the ones on the screen. The name of our ship is the Donovan Jones. The other giant bodies are other ships with crews of their own and captains, of course. I am going to hail them.

"But Sir, how can we deduce such a thing? Would it be wise to..."

Ignoring his navigator, Donimunt picked up the microphone attached to the command chair. His voice sounded strangely deep as it re-entered the control room through the loudspeakers:

"Ahoy, the giants. This is Donimunt, captain of the Donovan Jones. We are marooned. I would speak with one of your captains.

"Ooh, we're going to speak to the captains." Fuzbuji was beside himself with delight. But the others were too busy watching the puzzled faces of the giants. One of the giants spoke to the other:

"What did he just say?"

"Something about captains. He's still out of it. But he knows his name."

"You've had quite an accident, Mr. Jones. Just relax. Your partner, Mr. Melnick, came to see you. He'll be back later".

The captains of the giant-ships had refused to acknowledge Donimunt's message. This only increased his determination to leave this horrible place. The sooner the better. He turned once again to Brunt and Sorebrum.

"I want you to start this ship up. Now!"

As one of the giants on the screen worked on their ship from the outside, Sorebrum and Brunt bent over the instrument panel, on the inside. They tried one control after another. None of the switches had any detectable effect on the ship. Then they started to experiment with a bank of levers.

Suddenly a giant arm flew across the screen and struck something. A meaty thud echoed through the control room. The giant that had been working on their ship groaned and disappeared from view.

Donimunt was furious.

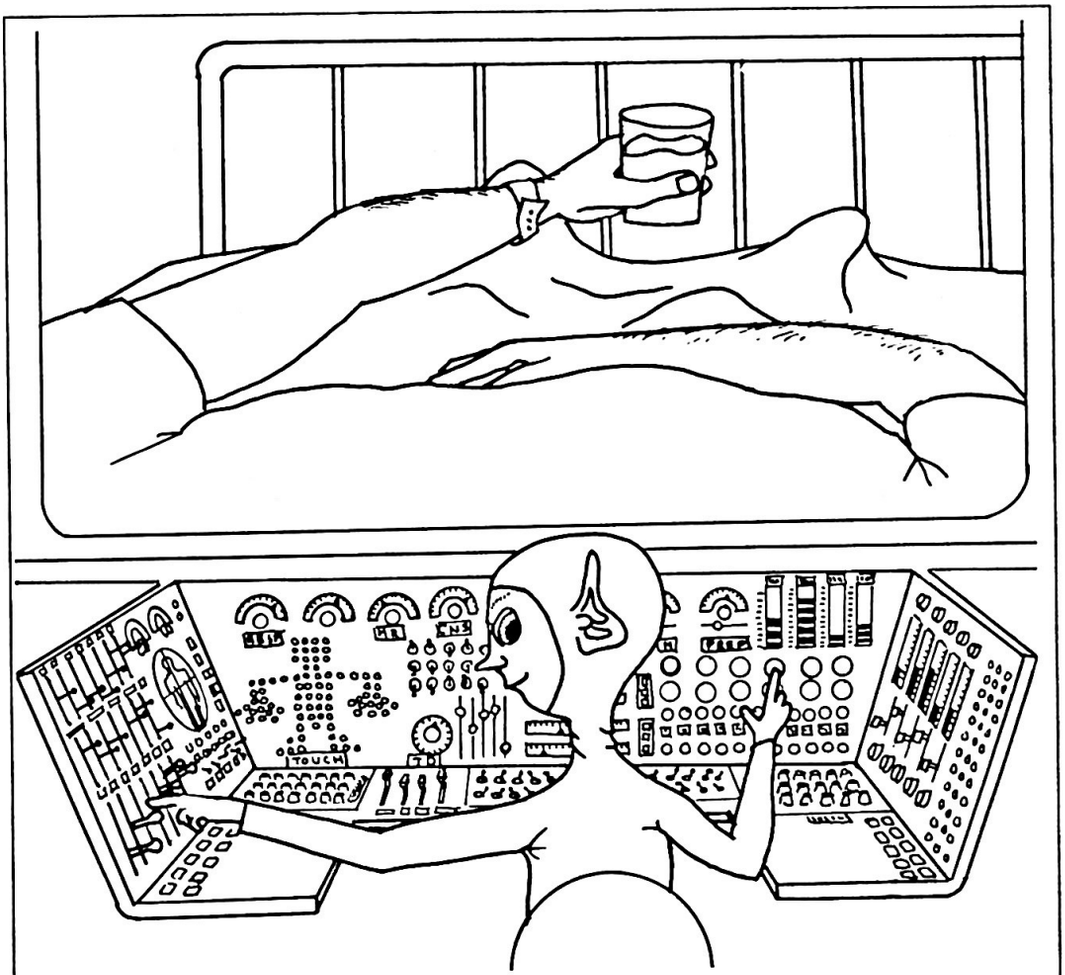
"What was that? You call yourself a pilot? Xyla could do better."

The captain referred to their passenger, Xyla of the Royal Brood, entrusted to their care for the journey to Modus Ponens and now shipwrecked with them.

Sorebrum nervously ran a hand over his enormous cranium as he pointed to the lever.

"It appears that this one controls the, ah, "arm" of our new ship!"

Over the days that followed, Brunt became increasingly proficient at operating the giant body-ship. By trial and error, he and Sorebrum pieced together the various functions of the control panel. Sorebrum even labelled the switches, levers and dials.



"Over the days that followed, Brunt became increasingly proficient at operating the giant body-ship".

There was much else for the homunculids to learn in this strange new world. As Brunt took them for short walks along corridors, he would sometimes dock the Donovan Jones by a window at Donimunt's urging. There was obviously far more to the this level than they had imagined. It had a fixed and unvarying character that Donimunt had come to regard as somewhat soothing given all the uncertainties they faced.

The "doctors" as the white-coated giants were called, would smile at the sight of their ship.

"Doing much better I see, Mr. Jones."

"You will address me as Captain Donimunt."

They would sometimes take the Donovan Jones to a place called the Day Room. Here was a marvellous machine that each homunculid had come to appreciate in his (or her) own way. After Brunt had carefully parked the ship on a couch, he would turn the head of the Donovan Jones toward the television set and all the homunculids, including the captain, would sit back in their seats and watch the television on their own screen. Even more than Sorebrum, Listig the Librarian intently watched. He noted the behavior and habits of the giants. His vast, encyclopedic brain took in baseball games, talk shows, newscasts and even daytime serials.

The serials held a special interest for Fuzbuji, not to mention Graspr, the Hedonics Officer. Being exceptionally ugly himself, he found the giants, especially the females, not unattractive. Occasionally a male and female giant would put their mouths together. Graspr would lick his fleshy lips and moan. Fuzbuji would merely sigh.

One day, their doctor introduced the Donovan Jones to a new giant. The psychiatrist Delbert Broome had a florid face and a hearty manner. He remarked that, physically speaking, their body was in great shape. Donimunt asked whether they could now leave, their ship being fully repaired. Broome explained that they would first have to move to a new ward where he would observe them for a week or so.

Later that day, Dr. Broome conducted an initial interview with his new patient. Donimunt was pleased that at last one of the giants acknowledged his presence. Dr. Broome was pleased to find a rare and controversial syndrome on his very doorstep. At length he made the following pronouncement:

"I think there's no harm in telling you that I know of cases like yours. The problem is called multiple personality disorder. Not that I personally regard it as a disorder, mind you. It's just a phase in your recovery from the crash. I feel quite confident that sooner or later there will be just one of you in there."

Donimunt was disturbed. All but one of them were destined to disappear. He took Sorebrum aside in confidence.

"Tell me, Navigator, what do you make of that statement by the captain of the Doctor Broome?"

"His statement is consistent with the theory that not all giants have crews. Those that do are regarded as having a "problem" or "disorder". Broome apparently proposes to eliminate all but one of us."

"I must warn you, Navigator, not to contradict my edicts. If I say that all giants have crews, then all giants have crews. As far as elimination is concerned, you are right. Lay plans for an immediate escape from the hospital. Say nothing to the others for now."

That night, when the psychiatric ward had grown quiet and the nurse had paid her last visit to the chamber where the Donovan Jones was berthed, Donimunt addressed his crew, fixing each in turn with a beady eye.

"Lads, tonight we make for the great outer world beyond the hospital. From there we shall set a course for home. What do you say, Mister Fuzbuji, something to get us under way!"

Fuzbuji modified a song that he had used on other occasions. He sang in a fruity, tremulous voice:

"A capital ship for an outdoor trip
Is the excellent Donovan Jones
No concrete halls or hospital walls
Will hold its muscles and bones.

We'll soon be back on a logical track
Through the Ceteris Paribus Zones
Where Nebulous Thoughts make fuzzy dots
On the screen of the Donovan Jones."

No one quite knew what the song meant but it sounded rather exciting, especially to Fuzbuji. Donimunt was about to give his first order when all eyes turned to the back of the control room. There stood the Princess Xyla.

She had emerged from the homunculid apartments behind the control room, a creature of exquisite, elfin beauty. At once Graspr ran to kneel in front of her, slobbering on her hand. Fuzbuji said, "Ahh, Xyla," his eyes wet with emotion.

"Where are we?"

"We are about to begin a voyage, your highness," said Donimunt. It was unusual to hear him speak with such deference.

"May I watch?"

"Certainly. Listig will brief you on the situation."

Listig waddled to the row of seats at the back of the control room. He sat beside the princess and described the nature of their vessel and how one of the giants had proposed eliminating all but one of them.

On the other side of the princess sat Graspr, running a warty hand up and down Xyla's leg.

"Begone, you loathsome creature!"

Donimunt gave the order to cast off. Brunt began to flip switches and move levers. The screen suddenly lit with a soft glow to show the interior of their hospital room; the Donovan Jones had turned on the bed lamp. With only an occasional murmur of advice from the navigator, Brunt dressed the Donovan Jones and walked the body to the door. The homunculids watched as the door slowly opened to reveal a vista down the immense hospital corridor.

"All clear. Proceed at half speed. Navigator report."

Sorebrum spelled out the proposed course. It took them along the corridor, past the nursing station, around a corner and then down another corridor. At the end was a set of double doors that led out into the great world of trees, buildings and stars.

Fuzbuji sat in his chair and watched the corridor wall glide noiselessly by on the screen. The atmosphere of the control room seemed tense, as though disaster might erupt at any moment. He watched as Brunt eased the head of the Donovan Jones around a corner to reveal a female giant reading a magazine at the nursing station.

The next thing he knew, the wood grain of the nursing station counter slid past the view screen. Brunt was now crawling their ship behind the counter to escape detection by the giant nurse. Fuzbuji quivered with excitement at the idea.

"Well done, Brunt!"

"Silence in the control room," hissed Donimunt.

Once they had cleared the nursing station, Brunt got the Donovan Jones up on its legs once more. They made it all the way to the end of the second corridor. Here there was some trouble with the double doors.

Fuzbuji was the first to see the hospital orderly descending some stairs to their right. He was unusually large, even for a giant.

"Watch out!" he screamed.

"Action stations. Stand by for evasive action!" A klaxon horn sounded.

The orderly's voice came clearly over the loudspeaker:

"Hey. You're not supposed to be here!"

Fuzbuji sensed terrible danger. He began to scream hysterically, "We're all going to die. I just know it."

The Princess Xyla looked alarmed.

"SHUT UP" roared Donimunt.

Unfortunately the microphone had been left on. Donimunt's command re-entered the control room through the loudspeakers. The orderly froze in his tracks.

"Don't tell me to shut up, you son of a bitch!"

In short, Donimunt's first attempt to escape from the hospital did not turn out well. There was a fight. The Donovan Jones might even have won, had the Princess not intervened.

Among the few controls on Brunt's panel resembling those on the non-existent logic cruiser was a simple dial. Called the Time Dilator, it could speed up or slow down the action on the screen. It could not influence the course of events, however, except by giving the homunculids more time to think.

In preparing for battle Brunt had turned the time dilator all the way up to full. The orderly's charge had slowed to a dreamlike ballet.

A giant fist drifted slowly toward the control room until it nearly filled the screen. At the last minute, Donimunt ordered Brunt to duck their head. They watched the fist sail by the screen. There were large black hairs on the knuckles. Kung Fu by time dilation.

"Make a fist and fire on my command...Fire!"

Brunt manipulated the appropriate controls, bringing their own right hand into play. The homunculids watched the fist, speed to its target. The control room rocked slightly when it struck the orderly's forehead.

"Son-of-a-bitch." The orderly's voice was a low growl.

Now another fist came at them, this one from an unexpected direction. Brunt could not dodge it in time. Although the control room rocked and Fuzbuji cried out in fear, none of the homunculids felt any personal pain at the blow. As far as the Donovan Jones was concerned, pain was merely a pattern of lights on the control panel.

Suddenly Xyla stood up at the back of the control room.

"Stop this immediately," she cried, "This is ridiculous."

Donimunt whirled in his command chair, a look of anger clouding his features, even before the Princess.

"But your highness, this is war."

"Stop it immediately, I am sure the way home does not lie by this route. I won't have it."

Despairing, Donimunt turned once more to the screen. They seemed to be going down. The corridor tilted at a crazy angle and a thick hairy arm obscured the lower half of the screen. It was too late to do anything.

The post-paralytic accountant was taken back to his room and strapped to his bed. His eyes stared blankly at the ceiling in the semi-darkness.

Inside, the homunculids held council. The Princess had gone back to her apartment and Donimunt charged Sorebrum with laying yet a new course out of the hospital. In

fact, Sorebrum already had a plan, but presenting it required the greatest possible tact. Donimunt had been getting increasingly touchy of late.

"By your leave glorious Captain, I will call on Officer Listig to recite the text of one of the magazine articles we happened to read in the day room recently. It describes a giant suffering from Multiple Personality Disorder."

Listig began. His voice sounded wheezy and his breath came in gasps.

"Cathy was a typical Multiple Personality patient. When her several selves at last came to know each other and the dominant personality took a leading role, she was already well on her way to a cure. In the weeks that followed, Holly the Hooker and Susan the Shy Girl, even Katherine the Actress manifested less and less frequently. Eventually they disappeared altogether. Today Cathy is a successful legal secretary, a responsible and happy member of her community. Asked if she would ever..."

"What's the point of this endless ramble," demanded Donimunt.

"Precisely this, Sir. It's a recipe for escape. It appears that the simplest way to get out of the hospital is to be cured. But to be cured in our case means that we must show no more symptoms of what the giants call Multiple Personality Disorder. In short we must pretend that there is but one entity in control of the Donovan Jones and that entity must call itself Donovan Jones. It's perfectly simple".

"And perfectly stupid," growled Donimunt, "What if they start eliminating us first?"

The issue remained unresolved. Donimunt ordered the crew to stand down and retire.

While the others slept, Sorebrum stayed up with Listig, going over magazine articles and television programs that the Librarian had recorded. There were numerous examples of phrases and gestures that the giants used but Sorebrum soon came to realize that to imitate a giant named Donovan Jones it would be far more useful to learn some of the details of the life they had interrupted.

The very next morning, Doctor Broome met his new patient again. Broome had high hopes for Donovan Jones. The case was unusual enough to merit more than a few pages in the International Psychiatric Journal. Was Donovan's attempt to escape the hospital last night a cry for help or a signal that changes were in the offing? The therapy would clearly require every tool in Doctor Broome's psychiatric kit.

Donovan Jones entered the office and sat carefully in the chair in front of Broome. He disconcerted Broome by sitting perfectly still and staring fixedly at the psychiatrist's right ear, as though a fly were perched there.

"Good morning to the Donovan Jones. Is Captain Donimunt there?"

"Of course I am. Is that the captain of the Doctor Broome?"

"What makes you think I have a captain and crew?"

"Who am I speaking to if not the captain? You sound like a captain. Perhaps you have no crew but I don't see how you can run such an enormous ship by yourself."

Suddenly Broome thought he saw an opening.

"I suppose you *could* describe me as a captain with no crew. Yes, it's tough being the captain of a giant body with no crew to help out. Tell me about your father."

"I was willed into being by the common imagination many eons ago."

By the end of the interview, Dr. Broome found himself adrift on a sea of new information. He ate lunch in his office, listening to the interview tape and making copious notes. Gradually a scheme suggested itself.

For their part, the homunculids took the Donovan Jones to the day room. Here, Donimunt ordered Brunt to park their ship on a couch by the window. Leaning back in his command chair he remarked somewhat petulantly that none of the giants would admit to having a crew. Even the psychiatrist would barely admit it. Sorebrum suggested that the problem was caused by the class of giants called Doctors. The patients might be more forthcoming.

Donimunt put the ship on a conversational footing and turned to the giant who sat on the same couch reading a magazine. This was a young male with long, shoulder-length hair. He slowly dropped the magazine, staring at the Donovan Jones with ever-increasing wonder as Donimunt explained that they were a shipwrecked crew of homunculids who had taken over a giant's body. he drew back uncomfortably when the other patient demanded that his own captain address him.

"Hey man, I don't have a captain. None of us does."

"You certainly do!"

"If you think everybody has a captain, what about the guy you took over? Did he have one?"

A long pause ensued in the conversation. The control room hung in a heavy air of indecision. Donimunt turned to Listig.

"You, Listig, what crew did we find here?"

"Er, none Sir."

Sorebrum smiled. He might even have pressed the point but they were suddenly interrupted by a nurse.

"Mr. Jones, a Robert Melnick is

here to see you."

A rotund gentleman in an expensive pinstripe suit entered the day room, looking confusedly about until he spotted his partner.

"Donovan! Donovan! It's so great to see you. I tried to visit you last week. We've all been so worried."

Donovan Jones merely stared uncomprehendingly at his visitor. Behind the expressionless eyes, Donimunt sat in his command chair trying to decide what to do next. The Princess Xyla had entered the Control Room once again, much to his annoyance. Then Sorebrum spoke:

"Here on our screen, oh esteemed Captain, is one who might provide much useful information for a voyage into the outer world. If you but command me to speak with him, pretending to be the giant called Donovan Jones, I might demonstrate the general technique and, at the same time, acquire the information."

It was a pretty speech, as much designed to impress the Princess as to persuade Donimunt. And when Xyla, by a wave of her Royal hand, tokened assent, it was as good as a command. Reluctantly, Donimunt handed the microphone over.

"Hello, Bob. As you can see, I'm not in the, er, greatest shape."

"Donovan, it's just so good to hear you speak."

"You've got to help me get my memory back."

"Certainly, Donovan, anything you say."

While Listig carefully recorded all the details, Melnick told the unseen audience everything he knew about the childhood, youth and adult years of Donovan Jones. The homunculids, especially Fuzbuji, were spellbound. Their ship had once been highly regarded in the local business community, whatever that was. When the nurse finally interrupted, Sorebrum pressed his plan once again.

It irritated Donimunt mightily that the Princess nodded as Sorebrum explained the details of their "cure."

"But I am the Captain of this vessel and I must remain the captain in all circumstances." Donimunt's dark eyes flashed dangerously.

At this, Xyla spoke up in a silvery voice.

"It seems the Navigator Sorebrum has plotted a wise course for our vessel. And is it not a course, after all, like any other? By the authority of the Royal Brood, I will ensure that you, my faithful Donimunt, will be in control at all times." She permitted the rarest and most precious of smiles to cross her features. Fuzbuji fainted with delight and even the captain's heart, hardened to its need for supremacy, felt some comfort.

Doctor Broome's heavy case load permitted him to see most patients once a week, only. But his newest case required close supervision. That is why, on the very next morning, Donovan Jones sat in front of him once again.

"How are we this morning?"

"I'm fine, thank you."

"And who am I speaking to this morning?"

"Donimunt, of course."

"Of course. I thought perhaps I would speak with someone else in there to-day."

"I'm afraid that won't be possible."

"And why not?"

"My crew died last night. One by one, they just dropped dead."

Dr. Broome experienced a deep ambivalence at the news. he frowned, smiled, then bit his lip.

"All dead, you say? How do you feel about being a captain with no crew?"

"I can manage. It helps now that certain memories are coming back."

"What memories?"

"I can remember stealing apples with Bob Melnick when I was ten."

The rehabilitation of Donovan Jones began in earnest from this point. Needless to say, Doctor Broome did not find his patient's new found memory entirely convincing. In fact he felt bitter. A strange twist of fate had cheated him of a miraculous cure. In the very next session Donimunt suggested that Dr. Broome address him as "Donovan." A flood of more memories followed. Donovan Jones stayed in the hospital another week but the psychiatric board had no further option but to release him.

That Friday, a lean, middle-aged man with hazel eyes walked confidently down the long front walk of the hospital toward a waiting car. Inside the graying temples everything was as normal as could be:

Donimunt sat in his command chair.

"Steady as she goes. Keep the head back. Eyes forward."

Brunt's hands flew expertly over the control panel. Sorebrum pondered vague new plans while Listig noted the newly budding trees that lined the walk. Fuzbuji watched the approaching car, and trembled with anticipation. Graspr eyed the Princess Xyla and the Princess, all alone, wondered if they would ever get to Modus Ponens.

Next issue:
the sequal to Alphie and Omega.

BULLETIN BOARD

Software and publications available from individuals and small companies.

EXCEL TEMPLATES AND ORIGINAL MACINTOSH ARTWORK

- LOTTO ANALYSIS
- WORLD DYNAMICS
- COVERSATION FACTORS
- PORTRAITS VOL 1
- PORTRAITS VOL 2

SEND $5.00 PER DISK TO COVER POSTAGE AND HANDLING TO:

RONALD A. OLANSEN
MACINTOSH ROAD, LYME, CT. 06371

***** 'I Owe Plenty' ---'Mark III' *****

Gives Amortization Schedules of Loans & Extra Payment/ Other Investment Advice. For PC Compaibles. Most versatile and crystal clear. Easily shows *huge saving* usually possible.

Pleased or money back, for only $18.00 (in Hawaii add 4%)

Joe M. Jeffers
92-1041 Makakilo Drive
#83, Ewa Beach, HI USA 96707

With bonus, latest TOPO set of Seven Games.
(State diskette size wanted)

DISCOVER CROSS-EYED 3-D:

If you can cross your eyes, you can see and manipulate *true* 3-dimentional split-image wire models of polyhedra, and nest them in the manner of M.C. Escher. For IBM compatible w/EGA or Hercules. Basic included. $5.00 postpaid.

1990 FRACTAL CALENDAR:

Thirteen *novel* laser-printed monochrome fractal images, including perspective and interior maps of the mandelbrot set. $5.00 postpaid.

J.E. Loyless 5185 Ashford Ct. Lilburn, GA 30247

Whether or not you have or even want the book, you may like to have the (ms-dos basica) programs in it:

Lessons in Edgeworth's Box	graphics of General Equilibrium
The Hyperbox	m goods and n traders
LP Simplex Algorithm	specially for class excerise
Power Algorithm for Minimum Paths	PROD=SUM, SUM=MIN
Von Neumann's Economic Model	the Stretch algorithm
Two Person Zero-sum Matrix Games	Minimaz
Stochastic Matrix Powers	contraction map graphics
Shao Yung's Program ?! & c...	...And a few additional items, some related to Kee Dewdney's articles in *Scientific American*.

Logic of Choice and Economic Theory

Oxford: Claredon Press, 1987 — Review in Economic Journal, March 1989

S. N. Afriat — 452 Roxborough Ave, Ottawa K1M 0L2 — 5 1/4" OR 3 1/2", US $15

T1-74 & TI-95 SUPPORT

A quarterly newsletter supports hand-held computers and programmable calculators including the TI-59, TI-74, TI-95 and CC-40.

For a sample copy send one dollar to:

TI PPC NOTES,
P.O. Box 1421, Largo FL 34649.
No checks please.

Loglan 1: **A Logical Language**
4th ed., 599 pp, $24. postpaid

A computer-intelligible scientific language, able to incorporate the vocabulary of any science. Teaching software, cassettes, parsers, dictionaries, workshops available.

The Loglan Institute
1701 NE 75th St., Gainesville, FL 32601
(904) 371-9574

What is the APL Programming Language?

APL is a quick and efficient tool for exploratory numerical analyses of Algorith puzzles/articles.

Call the BBS/APL 301-384-3672 3/2/2400 baud, 24 hrs. Many downloadable files (IBM/DOS, not Apple), including 2 free APL interpreters.

Queleques fiches et menus en françaiase.

Hallucinations™ 2.0

Screen Pattern Generator for IBM and Compatibles

This package consists of eighteen nonrepeating, menu-selectable graphics routines. CGA, EGA, and VGA graphics are supported. Many parameters can be altered from the Options menu or during execution. $20.00. Please add $2.00 shipping and handling per order. (California residents, add applicable sales tax.) Specify 5-¼" or 3½" diskette.

"Everyone with a color monitor should have a copy of this program." --Ric Manning, *The Louisville Times*.

Polymath Systems
P.O. Box 795, Berkeley, CA 94701

High-Range I.Q. Tests

(Ceiling Approximately 180)

***Langdon Adult Intelligence Test*, published in *Omni* (1979), scored by Polymath Systems; *Polymath Intellectual Ability Scale*, published in *Games* (1987), self-scoring. $10.00 each on paper (scoring included with *LAIT*); $10.00 for both on IBM-format diskette (*LAIT* scoring $10.00 additional). Please add $2.00 shipping and handling per order. (California residents, add applicable sales tax.) Specify 5-¼" or 3-½" diskette.**

Polymath Systems
P.O. Box 795, Berkeley, CA 94701

Hypertext Authors Wanted

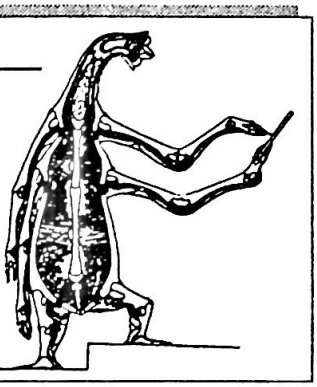
Please send proposals to:

Hypertext
corporation

63-36 98th Place, Suite 5J
Rego Park, NY 11374

YENDRED LIVES!

Follow Yendred, the Two-dimensional being, as he explores Arcle and the Planiverse. Copies of *The Planiverse* available from A.K. Dewdney. Send $10.00 U.S. (or equivalent) to the author at 42 Askin Street, London, Ontario CANADA N6C 4E4
Overseas orders add $2.00 U.S. for shipping.



VGAMBROT 1.04

There may be other Mandelbrot/Julia programs for the PC, but this is

THE FAST ONE

Reg'd: VGA or EGA Recc: 80x87, mouse
Send $30.00 to M. Freeman, (604) 988-2822
4777 Hoskins Road, N. Vancouver, B.C. V7K 2R3

CRYPTOLOGIA, founded in 1977, is a quarterly journal devoted to all aspects of cryptology including computer security, history, cipher machines, reviews, encryption schemes, memoirs, cryptanalytic techniques and much more...
Write for free Table of Contents for 1977-1988 issues and subscription information to:
Dr. Brian Winkel, Cryptologia,
Rose-Hulman Institute of Technology,
Terre Haute, IN 47803.

Join the ICWS and play Core War with the Worlds Best Programmers!

The Benefits:

1. an entry to the preliminary rounds of the annual ICWS tournament, and a chance at the prize.
2. a one year subscription to The Core War Newsletter.
3. a copy of the current standard, CWS'88.
4. The development of programming skills normally atrophied by typical corporate applications.
5. combat versus warriors from the world over!

Send $25.00 (check or M.O. drawn on U.S. Bank only)
to ICWS, Office of the Secretary
5717 Kern Drive, Huntington Beach, CA 92649-4535

NEED CORE WAR SOFTWARE? WANT THE BEST!?

If you have an IBM PC or compatible, at least 256K of RAM, and a CGA Display (or emulator), AMRAN has the Core War Software of you dreams!

For prompt delivery, send $34.95 U.S. to:
AMRAN, 5712 Kern Drive
Huntington Beach, CA 92649-4535
Core War Coliseum, Recode Assembler, sample Warrior Programs, and 40 page manual included on single 360K floppy disk.

TRS 80/IV with home-cooked BASIC programs incompatible with any current DOS or hardware, seeks fellow victim to exchange programs -- accounting, rosters, label printing, form letters, math, teaching, and games -- none boughten. Modem inop.

H.R. Catherwood, 303 777 6730,
130 Lafayette Street, Denver, CO 80218

1/2 PRICE FOR ALGORITHM READERS

Simulated Evolution. Not a single line of code tells the bugs how to solve their problems. See Scientific American, May '89: ~~$39.95~~ 19.90

Stroop Color Conflict Effect	~~$40.00~~	20.00
Piaget's Cognitive Operations (test for)	~~$40.00~~	20.00
Analysis of Extrasensory Perception	~~$40.00~~	20.00

For IBM-PC & compatibles, CGA, 256K. Add $3.00 S&H in cont. U.S., $4.00 Canada (U.S. funds on bank in the U.S. or Canadian Postal Money Order in U.S. funds).
Write for catalog:
Life Science Associates One Fenimore Road, Bayport, NY 11705

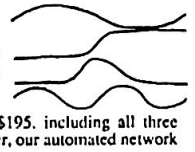
New! BrainMaker V2.0

The #1 selling Neural Network.

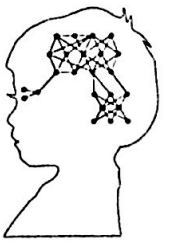
Over 7,000 copies sold. Here's why:

Best documentation: 255 pg. *Introduction to Neural Networks*. 332 pg. Users Guide. 80 pg. Application Guide. Our documentation includes everything you need to know to build networks and applications now.
Easiest to use: Pull down menus and dialog boxes. Supports mouse, color. NetMaker automatic network generation program helps you build networks in seconds.
Fastest:
128,000 cps on 286/8;
250,000 cps on 285/20;
500,000 cps on 386/33;
3 Million cps on C25;
4 Billion cps on N64;

• BrainMaker is $195. including all three books and Netmaker, our automated network generation program.



"BrainMaker is the most fascinating computer software I've ever seen. This is hotter than hot."
John Dvorak
PC Magazine



Call or write for a Free Catalogue!
California Scientific Software
160 Montecito #E Sierra Madre, CA 91024
(818) 355-1094

Recreational & Educational Computing

The only publication devoted to the playful connection of computers and math, The REC Newsletter features programming, fun, and challenges:
The magic of Schram's 123 String; the probability of an N-game at bingo; time to complete a set; the Conversational Computer; Newton's Method; Dr. Ecker's 'mathemagical' black holes; narcissistic numbers; 6174; Next Number in Sequence; Locate the Bomb; perfect numbers; Fibonacci numbers; prime number generation contest; self-reference and paradoxes; self-listing program; pi; probability; Monte Carlo simulations; fractal art; the world's best card trick; reviews of the best software and books; editorial; humor; cartoons; art; reader solutions.

SEND $24 for one year of 8 big issues...
By recreational computing columnist/math professor
Dr. Michael Ecker.
The REC Newsletter
129 Carol Drive, Clarks Summit, PA 18411
Make checks payable to REC

"The importance of the program cannot be overlooked... it so fundamentally alters the mechanics of mathematics."

New York Times

"Mathematica has quickly become a valuable tool in my daily work."

William Thurston
Professor of Mathematics,
Princeton University

"Mathematica has a real chance of replacing pencil and paper as the standard desktop environment for scientists and engineers."

William Press
Professor of Astrophysics,
Harvard University

1990 Mathematica Conference
January 11-13, 1990
Redwood City, California

Mathematica®

A System for Doing Mathematics by Computer

Some facts:

Function: Numerical, symbolic, graphical computation, interactive programming. Integrated technical computing environment.

Numerical Computation: Arbitrary precision arithmetic, complex numbers, special functions (hypergeometric, elliptic, etc), combinatorial and integer functions. Matrix operations, root finding, function fitting, Fourier transforms, numerical integration, minimization, linear programming.

```
In[1]:=
3^70
Out[1]=
2503155504993241601315571986085849
In[2]:=
Hypergeometric2F1[7,5,4.1,3-I]
Out[2]=
-0.00403761 - 0.00295663 I
```

Numerical Computation

Symbolic Computation: Equation solving, symbolic integration, differentiation, power series, limits. Algebraic operations, polynomial expansion, factorization, simplification. Operations on matrices, tensors, lists.

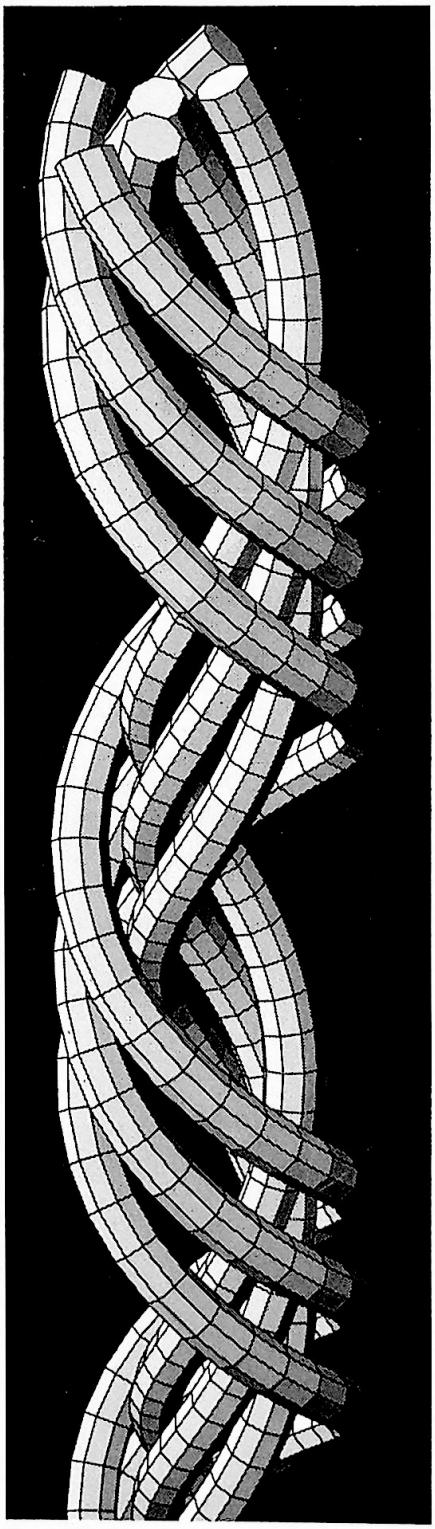
Graphics: 2D, 3D plots of functions, data, geometrical objects. Contour, density plots. 3D rendering with intersecting surfaces, lighting models, symbolic descriptions. Color PostScript output, publication quality graphics, animation (most versions).

```
In[1]:=
Integrate[x/(a + Exp[x]), x]
Out[1]=
                                  x
                                 E
  2      x Log[1 + ---]      PolyLog[2, -(---)]
 x         a                    a
---  - ------------ - ------------------
2a         a                  a
```

Symbolic Computation

Programming: High-level, interactive, symbolic system. Full procedural language, functional programming constructs. General transformation rule paradigm based on pattern matching.

External Interface: Input from external files, programs. Output in TeX, C, FORTRAN, PostScript. Interactive external process control (most versions).



Graphics and Visualization

Macintosh User Interface: Notebook interactive documents mixing text, graphics, animations, *Mathematica* input, output. Macintosh front end can be used with kernels on other computers. Macintosh graphics standards used.

Documentation: "*Mathematica*: A System for Doing Mathematics by Computer" by Stephen Wolfram (Addison-Wesley, 1988) available at bookstores. Additional documentation supplied with specific versions. *Mathematica* Journal to be published in 1990.

Versions Now Available: Apple Macintosh: $495 (Plus, SE, etc.); $795 (II, IIx, IIcx, SE/30, etc.) • 386-based MS-DOS systems: $695 (no coprocessor); $995 (287/387); $1295 (Weitek) • Apollo DN 3000, 4000: $2400 • DEC VAX VMS, ULTRIX, DECstation: from $2400 • Hewlett-Packard 9000/300, 800: from $2400 • IBM AIX/RT: $2400 • MIPS: from $2800 • NeXT: bundled as standard system software • Silicon Graphics IRIS-4D: from $2800 • Sony NEWS: from $2400 • Sun 3, 4, 386i: from $2250 • Supercomputer and other versions also available. • Educational, volume, reseller, and other discounts available • Now shipping Version 1.2.

```
log[1] = 0
log[E] = 1
log[x_ y_] := log[x] + log[y]
log[x_^n_] := n log[x]
log'[x_] = 1/x    (* derivative *)
log/: InverseFunction[log] = exp
log/:
Series[log[x_], {x_, 1, n_}] :=
 Sum[-(-1)^k (x-1)^k/k, {k,1,n}] +
                        0[x,1]^(n+1)
```

High-Level Programming

Implementation: 770 pre-defined *Mathematica* functions (C source 180,000 lines). Design, development led by Stephen Wolfram. Version 1.0 released June 1988.

Typical Applications: Research, engineering, education, mathematical modeling, publication graphics, data analysis, visualization, systems analysis, algorithm development.

Awards: Best 10 New Products, *Business Week* 1988 • Editor's Choice Award, *MacUser* 1989 • Award of Distinction, *BYTE* 1988.

Wolfram Research, Inc.

P.O. Box 6059, Champaign, IL 61826-6059, USA
Information: 217-398-0700. Orders: 800-441-MATH.
Or visit your local software dealer.

© 1989 Wolfram Research, Inc. *Mathematica* is a registered trademark of Wolfram Research, Inc. All other trademarks are trademarks of their respective companies. Prices and specifications are subject to change without notice. Graphic is a part of a tubular neighborhood of a six-strand braid generated by *Mathematica*.